Processamento de Linguagens e Compiladores

Trabalho Prático 1 - Exercíco 3

Carlos Beiramar (a84628) | Jorge Silva (a80931) | José Mendes (a81809)

(17 de novembro de 2021)

Resumo

O presente trabalho tem como objetivos, desenvolver a capacidade de criação de Expressões Regulares para descrição de padrões de frases dentro de textos, desenvolver *Filtros de Texto*, que filtrem ou transformem textos com base no conceito de regras de produção.

No presente relatório defendemos a nossa implementação em **Python** para o exercício 3, no qual é pretendido obter informação específica presente num ficheiro de texto.

Todos os resultados que obtivemos irão ser apresentados em forma de tabela num ficheiro *HTML* gerado para cada uma das soluções.

Conteúdo

1 Introdução

Na realização deste trabalho, optámos por implementar uma função em *Python* para resolver cada parâmetro do exercício 3. Inicialmente foi criado um menu para dar ao utilizador a escolher qual das informações é que este deseja ver. Assim, após a escolha do utilizador, usámos o módulo *re* do *Python* para filtrar o ficheiro de texto e, no final, irá ser criado um ficheiro *HTML* com uma tabela que irá apresentar toda a informação pedida.

2 Background

Este projeto está dividido em 2 ficheiros *Python*. O ficheiro **main.py** é o principal, onde todas as funções necessárias para a resolução de cada um dos parâmetros serão invocadas. Neste ficheiro está também presente a implementação da interface apresentada ao utilizador. Além deste, existe também o ficheiro **func.py** no qual está presente o código elaborado para a realização de cada um dos parâmetros pedidos no enunciado e, para além destas funções, foram criadas duas funções para gerar os ficheiros *HTML*.

3 Conceitos básicos

3.1 Ficheiro main

Funções implementadas:

- **init()** utiliza o módulo *re* do *Python* para criar um dicionário com toda a informação presente no ficheiro de texto (**info**). Após isto, chama uma função de acordo com o que o utilizador pretende ver.
- **printMenu()**, imprime o menu.

3.2 Ficheiro func

Funções implementadas:

- **genericHtml(headers,info,file)** e **genericHtmlList(headers,info,file)** ambas as funções permitem criar um ficheiro *html* no entanto, a segunda permite criar um ficheiro *html* quando, nos values no dicionário info, contém um lista com mais do que uma informação sobre os utilizadores.
- As funções seguintes foram criadas para solucionar os respetivos exercícios.
  - **ex1toHtml(info)**;
  - **ex2toHtml(info)**;
  - **ex3toHtml(info)**;
  - **ex4toHtml(info)**;
  - **ex5toHtml(info)**;
  - **ex6toHtml(info)**;
  - **toJson(info)**.
- Para além das funções referidas anteriormente, foi criada um função para criar uma tabela em *html* de toda a informação presente no ficheiro de texto.

4 A Proposta

De forma a simplificar o enunciado, o exercício 3 foi subdividido em vários exercícios.

4.1 Exercício 1

Aqui era pedido que se produzisse uma listagem apenas com o nome e a entidade do utilizador, ordenada alfabeticamente por nome. Esta informação está presente no ficheiro de texto auxiliar, *clav-users.txt*.

```
def ex1ToHtml(info):

    dic = {}

    for key,value in info.items():
        dic[value[0]] = value[1]

    dic = dict(sorted(dic.items(), key=lambda item: item[0]))
    genericHtml(["Nome", "Entidade"], dic, "ex1.html")
```

Figura 1: Código do exercício 1

4.2 Exercício 2

Neste exercício era pedido que fosse criada uma lista ordenada alfabeticamente das entidades referenciadas e, para cada uma delas, indicar quantos utilizadores estão registados.

```
def ex2ToHtml(info):

    dic = {}
    for value in info.values():
        if value[1] not in dic:
            dic[value[1]] = 1
        else:
            dic[value[1]] += 1
    dic = dict(sorted(dic.items(), key = lambda item: item[0]))

    genericHtml(["Entidade", "Quantidade"],dic,"ex2.html")
```

Figura 2: Código do exercício 2

4.3 Exercício 3

No exercício 3 era pedido que fosse mostrada a distribuição de utilizadores pelos níveis de acesso.

```
def ex3toHtml(info):

    dic = {}
    for key,value in info.items():
        if value[2] not in dic.keys():
            dic[value[2]] = []
            dic[value[2]].append(value[0])
        else:
            dic[value[2]].append(value[0])

    dic = dict(sorted(dic.items(), key = lambda item: item[0]))

    genericHtmlList(["Nível","Utilizadores"], dic, "ex3.html")
```

Figura 3: Código do exercício 3

4.4 Exercício 4

No presente exercício foi pedido que se produzisse uma listagem dos utilizadores, agrupados por entidade, ordenada primeiro pela entidade e dentro desta pelo nome.

```
def ex4toHtml(info):
    dic = {}

    for key,value in info.items():
        if value[1] not in dic.keys():
            dic[value[1]] = []
            dic[value[1]].append(value[0])
        else:
            dic[value[1]].append(value[0])

    for key in dic.keys():
        dic[key] = sorted(dic[key], key=str.lower)

    dic = dict(sorted(dic.items(), key = lambda item: item[0]))

    genericHtmlList(["Entidade", "Nome"], dic, "ex4.html")
```

Figura 4: Código do exercício 4

4.5 Exercício 5

Neste exercício era pedido que se apresentassem as soluções para as seguintes questões:

- Quanto utilizadores?
- Quantas entidades?
- Qual a distribuição em número por entidade?
- Qual a distribuição em número por nível?

Neste exercício em particular, a criação do ficheiro *HTML* é feita na própria função de forma a apresentar as soluções de todos os indicadores referidos anteriormente.

```
def ex5toHtml(info):
    dic_ent = {}
    dic_niv = {}

    for key,value in info.items():
        if value[2] not in dic_niv:
            dic_niv[value[2]] = 1
        else:
            dic_niv[value[2]] += 1

        if value[1] not in dic_ent:
            dic_ent[value[1]] = 1
        else:
            dic_ent[value[1]] += 1

    dic_ent = dict(sorted(dic_ent.items(), key = lambda item: item[1], reverse=True))
```

Figura 5: Código do exercício 5

4.6 Exercício 6

Foi acrescentado mais um exercício com intuito de mostrar toda a informação presente no ficheiro de texto numa tabela em *HTML*.

```
def ex6toHtml(info):
    lista=["Nome","Mail","Entidade","Nível","Chamadas backend"]
    info = dict(sorted(info.items(), key = lambda item:item[1]))

    f = open("html/ex6.html","w+")
    f.write("<!DOCTYPE html>\n")
    f.write("<html>\n")

    f.write("<style>\n")
    f.write("\ttable,td,th {\n \t\tborder: 1px solid black; \n")
    f.write("\t\tborder-collapse: collapse;\n \t\ttext-align: center;\n")
    f.write("\t\tmargin-right: 30px;\n\t}\n")
    f.write("</style>\n")

    f.write("\t<table>\n")
    f.write("\t\t<tr>\n") #uma linha da tabela
    for i in lista:
        f.write("\t\t\t<th>")
        f.write(i)
        f.write("</th>\n")
    f.write("\t\t</tr>\n")
    for key,value in info.items():
        f.write("\t\t<tr>\n")
        f.write("\t\t\t<td>")
        f.write(value[0])
        f.write("</td>\n")
        f.write("\t\t\t<td>")
        f.write(key)
        f.write("</td>\n")
        for i in range(1,len(value)):
            f.write("\t\t\t<td>")
            f.write(value[i])
            f.write("</td>\n")
        f.write("\t\t</tr>\n")
    f.write("\t</table>\n")
    f.write("</html>")
    f.close()
```

Figura 6: Código do exercício 6

4.7 Exercício 7

Aqui era pedido que fosse criado um ficheiro *Json* com os 20 primeiros registos do ficheiro de texto.

```
def toJson(info):
    reg = []
    counter = 0

    j = open('json/data.json', 'w+', encoding='utf-8')

    for key,value in info.items():
        l=[]
        l.append(key)
        for i in value:
            l.append(i)
        reg.append(l)

        counter+=1
        if counter == 20:
            break

    json.dump(reg, j, ensure_ascii=False, indent=4)
    j.close()
```

Figura 7: Código do exercício 7

5 Implementação

Em seguida serão explicadas todas as figuras apresentadas anteriormente de forma a explicar todo o procedimento estabelecido para resolver qualquer um dos parâmetros pedidos no enunciado. Inicialmente é criado um dicionário onde a *key* é o mail de cada um dos utilizadores e o *value* é uma lista com o nome, a entidade, o nível e o número de chamadas ao backend. Este dicionário será passado como parâmetro a todas as funções das imagens apresentadas anteriormente.

5.1 Figura 1

Na função **ex1ToHtml** é criado um dicionário auxiliar onde *key* será o nome do utilizador e o *value* será a respetiva entidade. De seguida, aplica-se um *sort* ao dicionário para o ordenar alfabeticamente pela *key*. No final é chamada a função **genericHtml** para criar a tabela no ficheiro *html*.

5.2 Figura 2

Na função **ex2ToHtml** cria-se um dicionário onde a *key* será cada uma das entidades existentes e o *value* será a quantidade de utilizadores que pertencem a essa entidade. No final é aplicado um *sort* ao dicionário para este ficar ordenado pelas entidades e é chamada a função **genericHtml** para criar a tabela no ficheiro *html*.

5.3 Figura 3

Na função **ex3ToHtml** é criado um dicionário onde cada *key* será um nível diferente e nos *values* de cada *key* está uma lista com o nome de cada um dos utilizadores que pertence a esse nível. Após isto, o dicionário é ordenado pela *key* e é chamada a função **genericHtmlList** para criar um ficheiro *html* com a respetiva tabela.

5.4 Figura 4

Nesta figura está implementada a função para solucionar o exercício 4. Inicialmente cria-se um dicionário onde as *keys* serão as entidades e os *values* serão listas com os nomes dos utilizadores que correspondem a esse entidade. Após isto, cada lista nos *values* é ordenada alfabeticamente e posteriormente, o dicionário é ordenado alfabeticamente pelas *keys*. No final é chamada a função **genericHtmlList** para criar um ficheiro com a respetiva tabela.

5.5 Figura 5

Nesta imagem encontra-se o código escrito para resolver o exercício 5. Aqui são criados 2 dicionários, **dic_ent** onde as *keys* são as entidades e os *values* são a quantidade de utilizadores que pertence a essa entidade, **dic_niv** onde as *keys* são todos os níveis existentes no ficheiro e os *values* a quantidade de utilizadores por nível. Apesar de não estar representado na imagem da figura 5, a criação do ficheiro *html* é feita dentro da função **ex5toHtml** de forma a ser possível imprimir todos os indicadores desejados num só ficheiro *html*.

5.6 Figura 6

Nesta figura é apresentado o código para imprimir a tabela que foi fornecida no ficheiro *txt*. Foi criada uma lista com os parâmetros presentes no ficheiro *txt*. De seguida, o dicionário com toda a informação do ficheiro foi ordenado alfabeticamente pelo nome dos utilizadores. A criação do ficheiro *html* é feita dentro da função **ex6toHtml** de forma a ser possível imprimir toda a informação.

5.7 Figura 7

Nesta imagem está presente o código implementado para listar os 20 primeiros utilizadores num novo ficheiro *json*. Inicialmente o ficheiro *json* é criado, após isso toda a informação dos utilizadores presente no dicionário **info** é copiada para uma nova lista pela seguinte ordem: email, nome, entidade, nível, chamadas ao backend. Após isto, toda a informação presente na lista vai ser imprimida num ficheiro *json*.

6 Código implementado

6.1 main.py

```
import re, func, os

def printMenu():

    print("\n \t\t\t\t\tMenu\n")
    print("\t[1] - Produz uma listagem apenas com o nome e a", end = "")
    print("entidade do utilizador.")
    print("\t[2] - Produz uma lista das entidades referenciadas,", end = "")
    print(" com o número de utilizadores.")
    print("\t[3] - Distribuição de utilizadores por níveis de acesso.")
    print("\t[4] - Listagem dos utilizadores, agrupados por entidade.")
    print("\t[5] - Indicadores:\n \t\t-> Quantos utilizadores?\n",end = "")
    print("\t\t-> Quantas entidades?\n", end="")
    print("\t\t-> Qual a distribuição em número por entidade?\n",end="")
    print("\t\t-> Qual a distribuição em número por nível?")
    print("\t[6] - Imprimir toda a informação.")
    print("\t[7] - Imprimir os 20 primeiros registos num novo ficheiro", end="")
    print(" de output em formato Json.")
    print("\t[0] - Sair")


def init():

    printMenu()
    val = input("Escolha uma opção:")
    f = open("clav-users.txt", "r")
    info = {}

    for line in f:
        string = re.sub(r"\n",r"",line)
        l = re.split(r'::',string)
        info[l[1]] = [l[0]] + l[2:]

    while val != 0:

        if val == '1':
            func.ex1ToHtml(info)
            print("Ficheiro html criado.")
            os.system("open -a \"Google Chrome\" html/ex1.html")

        elif val == '2':
            func.ex2ToHtml(info)
```

```
            print("Ficheiro html criado.")
            os.system("open -a \"Google Chrome\" html/ex2.html")

        elif val == '3':
            func.ex3toHtml(info)
            print("Ficheiro html criado.")
            os.system("open -a \"Google Chrome\" html/ex3.html")

        elif val == '4':
            func.ex4toHtml(info)
            print("Ficheiro html criado.")
            os.system("open -a \"Google Chrome\" html/ex4.html")

        elif val == '5':
            func.ex5toHtml(info)
            print("Ficheiro html criado.")
            os.system("open -a \"Google Chrome\" html/ex5.html")

        elif val == '6':
            func.ex6toHtml(info)
            print("Ficheiro html criado.")
            os.system("open -a \"Google Chrome\" html/ex6.html")

        elif val == '7':
            func.toJson(info)
            print("Ficheiro json criado.")
            os.system("open json/data.json")

        elif val == '0':
            print("Finalizado")
            os.system("rm html/* json/data.json")
            break
        else:
            print("Valor inválido.")
        print()
        printMenu()
        val = input("Escolha uma opção:")

    f.close()

init()
```

6.2 func.py

```
import json

def genericHtml(headers,info,file):

    f = open("html/" + file, "w+")
    f.write("<!DOCTYPE html>\n")
    f.write("<html>\n")

    f.write("<style>\n")
    f.write("\ttable,td,th {\n \t\tborder: 1px solid black; \n")
    f.write("\t\tborder-collapse: collapse;\n \t\ttext-align: center;\n \t}\n")
    f.write("</style>\n")

    f.write("\t<table>\n")
    f.write("\t\t<tr>\n")

    for i in range(0,len(headers)):
        f.write("\t\t\t<th>")
        f.write(headers[i])
        f.write("</th>\n")
    f.write("\t\t</tr>\n")

    for key,value in info.items():
        f.write("\t\t<tr>\n")
        f.write("\t\t\t<td>")
        f.write(key)
        f.write("</td>\n")
        f.write("\t\t\t<td>")
        f.write(str(value))
        f.write("</td>\n")
        f.write("\t\t</tr>\n")

    f.write("\t</table>\n")
    f.write("</html>")
    f.close()


def genericHtmlList(headers,info,file):
    f = open("html/" + file, "w+")
    f.write("<!DOCTYPE html>\n")
    f.write("<html>\n")

    f.write("<style>\n")
    f.write("\ttable,td,th {\n \t\tborder: 1px solid black; \n")
```

```
    f.write("\t\tborder-collapse: collapse;\n \t\ttext-align: center;\n \t}\n")
    f.write("</style>\n")

    f.write("\t<table>\n")
    f.write("\t\t<tr>\n")

    for i in range(0,len(headers)):
        f.write("\t\t\t<th>")
        f.write(headers[i])
        f.write("</th>\n")
    f.write("\t\t</tr>\n")

    for key,value in info.items():
        f.write("\t\t<tr>\n")
        f.write("\t\t\t<td>")
        f.write(key)
        f.write("</td>\n")
        f.write("\t\t\t<td>\n")
        for i in value:
            f.write("\t\t\t\t")
            f.write(i)
            f.write("<br>\n")
        f.write("\t\t\t</td>\n")
        f.write("\t\t</tr>\n")

    f.write("\t</table>\n")
    f.write("</html>")
    f.close()

def ex1ToHtml(info):

    dic = {}

    for key,value in info.items():
        dic[value[0]] = value[1]

    dic = dict(sorted(dic.items(), key=lambda item: item[0]))
    genericHtml(["Nome", "Entidade"], dic, "ex1.html")


def ex2ToHtml(info):

    dic = {}
    for value in info.values():
        if value[1] not in dic:
            dic[value[1]] = 1
```

```
            else:
                dic[value[1]] += 1
        dic = dict(sorted(dic.items(), key = lambda item: item[0]))

        genericHtml(["Entidade", "Quantidade"],dic,"ex2.html")

    def ex3toHtml(info):

        dic = {}
        for key,value in info.items():
            if value[2] not in dic.keys():
                dic[value[2]] = []
                dic[value[2]].append(value[0])
            else:
                dic[value[2]].append(value[0])

        dic = dict(sorted(dic.items(), key = lambda item: item[0]))

        genericHtmlList(["Nível","Utilizadores"], dic, "ex3.html")

    def ex4toHtml(info):
        dic = {}

        for key,value in info.items():
            if value[1] not in dic.keys():
                dic[value[1]] = []
                dic[value[1]].append(value[0])
            else:
                dic[value[1]].append(value[0])

        for key in dic.keys():
            dic[key] = sorted(dic[key], key=str.lower)

        dic = dict(sorted(dic.items(), key = lambda item: item[0]))

        genericHtmlList(["Entidade", "Nome"], dic, "ex4.html")

    def ex5toHtml(info):
        dic_ent = {}
        dic_niv = {}

        for key,value in info.items():
            if value[2] not in dic_niv:
                dic_niv[value[2]] = 1
            else:
                dic_niv[value[2]] += 1
```

```
    if value[1] not in dic_ent:
        dic_ent[value[1]] = 1
    else:
        dic_ent[value[1]] += 1

dic_ent = dict(sorted(dic_ent.items(), key = lambda item: item[1], reverse=True))

f = open("html/ex5.html","w+")
f.write("<!DOCTYPE html>\n")
f.write("<html>\n")

f.write("<style>\n")
f.write("\ttable,td,th {\n \t\tborder: 1px solid black; \n")
f.write("\t\tborder-collapse: collapse;\n \t\ttext-align: center;\n")
f.write("\t\tmargin-right: 30px;\n\t}\n")
f.write("\t.table-2{\n \t\theight:50%;\n\t}\n")
f.write("\tdiv {\n \t\tdisplay: flex;\n\t}\n")
f.write("\tspan {\n \t\tfont-weight: bold;\n\t}\n")
f.write("\tspan {\n \t\tfont-weight: bold;\n\t}\n")
f.write("</style>\n")

f.write("<p>")
f.write("<span>")
f.write("Total de utilizadores: ")
f.write("</span>")
f.write(str(len(info.keys())))
f.write("</p>\n")

f.write("<p>")
f.write("<span>")
f.write("Total de entidades: ")
f.write("</span>")
f.write(str(len(dic_ent.keys())))
f.write("</p>\n")
f.write("<div>\n")
f.write("\t<table>\n")
f.write("\t\t<tr>\n")

f.write("\t\t\t<th>")
f.write("Entidade")
f.write("</th>\n")
f.write("\t\t\t<th>")
f.write("Quantidade")
f.write("</th>\n")
```

```
for key,value in dic_ent.items():
    f.write("\t\t<tr>\n")
    f.write("\t\t\t<td>")
    f.write(key)
    f.write("</td>\n")
    f.write("\t\t\t<td>")
    f.write(str(value))
    f.write("</td>\n")
    f.write("\t\t</tr>\n")


f.write("\t</table>\n")

f.write("\t<table class = \"table-2\">\n")
f.write("\t\t<tr>\n")

f.write("\t\t\t<th>")
f.write("Nível")
f.write("</th>\n")
f.write("\t\t\t<th>")
f.write("Quantidade")
f.write("</th>\n")

for key,value in dic_niv.items():
    f.write("\t\t<tr>\n")
    f.write("\t\t\t<td>")
    f.write(key)
    f.write("</td>\n")
    f.write("\t\t\t<td>")
    f.write(str(value))
    f.write("</td>\n")
    f.write("\t\t</tr>\n")

f.write("\t</table>\n")
f.write("</div>\n")

f.write("</html>")
f.close()
```

```
def ex6toHtml(info):
    lista=["Nome","Mail","Entidade","Nível","Chamadas backend"]
    info = dict(sorted(info.items(), key = lambda item:item[1]))

    f = open("html/ex6.html","w+")
    f.write("<!DOCTYPE html>\n")
    f.write("<html>\n")

    f.write("<style>\n")
    f.write("\ttable,td,th {\n \t\tborder: 1px solid black; \n")
    f.write("\t\tborder-collapse: collapse;\n \t\ttext-align: center;\n")
    f.write("\t\tmargin-right: 30px;\n\t}\n")
    f.write("</style>\n")

    f.write("\t<table>\n")
    f.write("\t\t<tr>\n")
    for i in lista:
        f.write("\t\t\t<th>")
        f.write(i)
        f.write("</th>\n")
    f.write("\t\t</tr>\n")
    for key,value in info.items():
        f.write("\t\t<tr>\n")
        f.write("\t\t\t<td>")
        f.write(value[0])
        f.write("</td>\n")
        f.write("\t\t\t<td>")
        f.write(key)
        f.write("</td>\n")
        for i in range(1,len(value)):
            f.write("\t\t\t<td>")
            f.write(value[i])
            f.write("</td>\n")
        f.write("\t\t</tr>\n")
    f.write("\t</table>\n")
    f.write("</html>")
    f.close()
```

```
def toJson(info):
    reg = []
    counter = 0

    j = open('json/data.json', 'w+', encoding='utf-8')

    for key,value in info.items():
        l=[]
        l.append(key)
        for i in value:
            l.append(i)
        reg.append(l)

        counter+=1
        if counter == 20:
            break

    json.dump(reg, j, ensure_ascii=False, indent=4)
    j.close()
```

7 Conclusão

Após a apresentação do código implementado para cada um dos problemas pedidos e após a explicação individual de cada uma das funções implementadas, irão ser agora apresentados todos os resultados obtidos.
Lista dos resultados atingidos:

- Figura 1: resultado obtido

Nome	Entidade
Alda do Carmo Namora Soares de Andrade	ent_FLUL
Alexadre Teixeira	ent_KEEP
Alexandra Lourenço	ent_DGLAB
Alexandra Maria Alves Coutinho Rodrigues	ent_UTAD
Alexandra Testes	ent_A3ES
Alexandre Teixeira	ent_A3ES
Aluno de DAW2020	ent_A3ES
Ana Lúcia Cabrita Guerreiro	ent_CCDR-Alg
Ana Maria Teixeira Gaspar	ent_SGMF
António José Morim Brandão	ent_MdP
CLAV-migrator	ent_A3ES
Carlos Barbosa	ent_A3ES
Carlos Matoso	ent_IEFP
Cármen Isabel Amador Francisco	ent_CMSNS
Cátia João Matias Trindade	ent_DGLAB
Cátia Trindade	ent_DGLAB
DAW2020-teste	ent_A3ES
David Ferreira	ent_IEFP
Design-DGLAB	ent_DGLAB
Duarte Freitas	ent_A3ES
Fernando Manuel Antunes Marques da Silva	ent_STI-M
Filipa Carvalho	ent_DGLAB
Filipe Ferreira Cardoso Leitão	ent_CMSPS
Formação DGLAB	ent_DGLAB
Frederico Pinto	ent_ACSS
Joana Braga	ent_IEFP
José Carlos Leite Ramalho	ent_DGLAB
José Carlos Martins	ent_A3ES
João Paulo de Melo Esteves Pereira	ent_APA
João Pimentel	ent_A3ES
Madalena Ribeiro	ent_DGLAB
Manuel Monteiro	ent_A3ES
Maria Celeste Pereira	ent_DGLAB
Maria José Maciel Chaves	ent_DGLAB
Maria Leonor da Conceição Fresco Franco	ent_CCDR-LVT
Maria Matos de Almeida Talhada Correia	ent_ICNF
Maria Rita Gago	ent_DGLAB
Miguel Ferreira	ent_KEEP
Nuno Filipe Casas Novas	ent_CCDR-LVT
PRI2020-teste	ent_A3ES
Paulo Lima	ent_KEEP
Pedro Penteado	ent_DGLAB
Regina Neves Lopes	ent_SGMF
Ricardo Almeida	ent_DGEG
Ricardo Canela	ent_BdP
Rui Araújo	ent_II
Rui Araújo Entidade	ent_AAN
Rui Araújo Simples	ent_LNEC
Sandra Cristina Patrício da Silva	ent_CMSNS
Silvestre Lacerda	ent_DGLAB
Sónia Isabel Ferreira Gonçalves Negrão	ent_CMABF
Sónia Patrícia Pinheiro Reis	ent_ICNF
Zélia Gomes	ent_DGLAB
clara cristina rainho viegas	ent_DGLAB
jcm	ent_AAN
jcr-rep-entidade	ent_A3ES
octavio	ent_A3ES
Élia Cristina Viegas Pedro	ent_CCDR-Alg

Figura 8: Tabela exercício 1

- Figura 2: resultado obtido

Entidade	Quantidade
ent_A3ES	14
ent_AAN	2
ent_ACSS	1
ent_APA	1
ent_BdP	1
ent_CCDR-Alg	2
ent_CCDR-LVT	2
ent_CMABF	1
ent_CMSNS	2
ent_CMSPS	1
ent_DGEG	1
ent_DGLAB	16
ent_FLUL	1
ent_ICNF	2
ent_IEFP	3
ent_II	1
ent_KEEP	3
ent_LNEC	1
ent_MdP	1
ent_SGMF	2
ent_STI-M	1
ent_UTAD	1

Figura 9: Tabela exercício 2

- Figura 3: resultado obtido

Nível	Utilizadores
1	Élia Cristina Viegas Pedro Nuno Filipe Casas Novas Sónia Patrícia Pinheiro Reis Sónia Isabel Ferreira Gonçalves Negrão Filipe Ferreira Cardoso Leitão Ana Lúcia Cabrita Guerreiro Alda do Carmo Namora Soares de Andrade Ricardo Almeida Sandra Cristina Patrício da Silva Fernando Manuel Antunes Marques da Silva Ana Maria Teixeira Gaspar Maria Matos de Almeida Talhada Correia Cármen Isabel Amador Francisco Maria Leonor da Conceição Fresco Franco Regina Neves Lopes João Paulo de Melo Esteves Pereira António José Morim Brandão Rui Araújo Simples Rui Araújo Entidade jcm Alexandra Testes Alexandra Maria Alves Coutinho Rodrigues jcr-rep-entidade
2	octavio Aluno de DAW2020 PRI2020-teste DAW2020-teste Carlos Matoso Joana Braga David Ferreira
3	Ricardo Canela
3.5	Formação DGLAB
4	Cátia João Matias Trindade Cátia Trindade Filipa Carvalho clara cristina rainho viegas Zélia Gomes Madalena Ribeiro Maria Celeste Pereira Maria José Maciel Chaves
5	Pedro Penteado Silvestre Lacerda
6	Maria Rita Gago Madalena Ribeiro
7	Miguel Ferreira José Carlos Leite Ramalho Alexandra Lourenço Rui Araújo Frederico Pinto Alexandre Teixeira José Carlos Martins Design-DGLAB João Pimentel CLAV-migrator Alexadre Teixeira Carlos Barbosa Manuel Monteiro Duarte Freitas José Carlos Leite Ramalho Paulo Lima

Figura 10: Tabela exercício 3

- Figura 4: resultado obtido

Entidade	Nome
ent_A3ES	Alexandra Testes Alexandre Teixeira Aluno de DAW2020 Carlos Barbosa CLAV-migrator DAW2020-teste Duarte Freitas jcr-rep-entidade José Carlos Leite Ramalho José Carlos Martins João Pimentel Manuel Monteiro octavio PRI2020-teste
ent_AAN	jcm Rui Araújo Entidade
ent_ACSS	Frederico Pinto
ent_APA	João Paulo de Melo Esteves Pereira
ent_BdP	Ricardo Canela
ent_CCDR-Alg	Ana Lúcia Cabrita Guerreiro Élia Cristina Viegas Pedro
ent_CCDR-LVT	Maria Leonor da Conceição Fresco Franco Nuno Filipe Casas Novas
ent_CMABF	Sónia Isabel Ferreira Gonçalves Negrão
ent_CMSNS	Cármen Isabel Amador Francisco Sandra Cristina Patrício da Silva
ent_CMSPS	Filipe Ferreira Cardoso Leitão
ent_DGEG	Ricardo Almeida
ent_DGLAB	Alexandra Lourenço clara cristina rainho viegas Cátia João Matias Trindade Cátia Trindade Design-DGLAB Filipa Carvalho Formação DGLAB José Carlos Leite Ramalho Madalena Ribeiro Madalena Ribeiro Maria Celeste Pereira Maria José Maciel Chaves Maria Rita Gago Pedro Penteado Silvestre Lacerda Zélia Gomes
ent_FLUL	Alda do Carmo Namora Soares de Andrade
ent_ICNF	Maria Matos de Almeida Talhada Correia Sónia Patrícia Pinheiro Reis
ent_IEFP	Carlos Matoso David Ferreira Joana Braga
ent_II	Rui Araújo
ent_KEEP	Alexadre Teixeira Miguel Ferreira Paulo Lima
ent_LNEC	Rui Araújo Simples
ent_MdP	António José Morim Brandão
ent_SGMF	Ana Maria Teixeira Gaspar Regina Neves Lopes
ent_STI-M	Fernando Manuel Antunes Marques da Silva
ent_UTAD	Alexandra Maria Alves Coutinho Rodrigues

Figura 11: Tabela exercício 4

- Figura 5: resultado obtido

Total de utilizadores: 60

Total de entidades: 22

Entidade	Quantidade
ent_DGLAB	16
ent_A3ES	14
ent_KEEP	3
ent_IEFP	3
ent_CCDR-Alg	2
ent_CCDR-LVT	2
ent_ICNF	2
ent_CMSNS	2
ent_SGMF	2
ent_AAN	2
ent_CMABF	1
ent_CMSPS	1
ent_FLUL	1
ent_DGEG	1
ent_BdP	1
ent_STI-M	1
ent_APA	1
ent_MdP	1
ent_II	1
ent_ACSS	1
ent_LNEC	1
ent_UTAD	1

Nível	Quantidade
1	23
3.5	1
4	8
3	1
7	16
6	2
5	2
2	7

Figura 12: Indicadores exercício 5

- Figura 6: resultado obtido

Nome	Mail	Entidade	Nível	Chamadas backend
Alda do Carmo Namora Soares de Andrade	aandrade@letras.ulisboa.pt	ent_FLUL	1	0
Alexadre Teixeira	alex@keep.pt	ent_KEEP	7	0
Alexandra Lourenço	alexandra.lourenco@dglab.gov.pt	ent_DGLAB	7	3
Alexandra Maria Alves Coutinho Rodrigues	acoutinh@utad.pt	ent_UTAD	1	0
Alexandra Testes	m-alexandra.lourenco@dglab.gov.pt	ent_A3ES	1	0
Alexandre Teixeira	a73547@alunos.uminho.pt	ent_A3ES	7	0
Aluno de DAW2020	leo.ramalho99@gmail.com	ent_A3ES	2	0
Ana Lúcia Cabrita Guerreiro	alucia@ccdr-alg.pt	ent_CCDR-Alg	1	0
Ana Maria Teixeira Gaspar	ana.gaspar@sgmf.gov.pt	ent_SGMF	1	0
António José Morim Brandão	Antonio.brandao@metrodoporto.pt	ent_MdP	1	0
CLAV-migrator	a74036@alunos.uminho.pt	ent_A3ES	7	0
Carlos Barbosa	A82324@alunos.uminho.pt	ent_A3ES	7	0
Carlos Matoso	cmatoso@ambisig.com	ent_IEFP	2	0
Cármen Isabel Amador Francisco	carmem@mun-sines.pt	ent_CMSNS	1	0
Cátia João Matias Trindade	catia.trindade@dglab.gov.pt	ent_DGLAB	4	0
Cátia Trindade	matiasjcatia@gmail.com	ent_DGLAB	4	0
DAW2020-teste	daw2020@teste.uminho.pt	ent_A3ES	2	0
David Ferreira	david.ferreira@ambisig.com	ent_IEFP	2	0
Design-DGLAB	a75536@alunos.uminho.pt	ent_DGLAB	7	0
Duarte Freitas	a63129@alunos.uminho.pt	ent_A3ES	7	0
Fernando Manuel Antunes Marques da Silva	fernando.marques.silva@marinha.pt	ent_STI-M	1	0
Filipa Carvalho	filipa.carvalho@dglab.gov.pt	ent_DGLAB	4	0
Filipe Ferreira Cardoso Leitão	arquivo@cm-spsul.pt	ent_CMSPS	1	0
Formação DGLAB	lurdes.almeida@dglab.gov.pt	ent_DGLAB	3.5	0
Frederico Pinto	frederico21pinto@hotmail.com	ent_ACSS	7	0
Joana Braga	jbraga@ambisig.com	ent_IEFP	2	0
José Carlos Leite Ramalho	jcr@di.uminho.pt	ent_A3ES	7	3
José Carlos Leite Ramalho	jcr@dglab.pt	ent_DGLAB	7	0
José Carlos Martins	a78821@alunos.uminho.pt	ent_A3ES	7	0
João Paulo de Melo Esteves Pereira	joao.pereira@apambiente.pt	ent_APA	1	0
João Pimentel	A80874@alunos.uminho.pt	ent_A3ES	7	0
Madalena Ribeiro	madalena.ribeiro07@gmail.com	ent_DGLAB	4	2
Madalena Ribeiro	madalena.ribeiro@dglab.gov.pt	ent_DGLAB	6	0
Manuel Monteiro	a74036@alunos.uminhopt	ent_A3ES	7	0
Maria Celeste Pereira	m-celeste.pereira@dglab.gov.pt	ent_DGLAB	4	0

Figura 13: Tabela exercício 6

Maria José Maciel Chaves	m-jose.chaves@dglab.gov.pt	ent_DGLAB	4	0
Maria Leonor da Conceição Fresco Franco	leonor.mina@ccdr-lvt.pt	ent_CCDR-LVT	1	0
Maria Matos de Almeida Talhada Correia	MariaMatos.Correia@icnf.pt	ent_ICNF	1	0
Maria Rita Gago	m-rita.gago@dglab.gov.pt	ent_DGLAB	6	3
Miguel Ferreira	mferreira@keep.pt	ent_KEEP	7	0
Nuno Filipe Casas Novas	nuno.novas@ccdr-lvt.pt	ent_CCDR-LVT	1	0
PRI2020-teste	pri2020@teste.uminho.pt	ent_A3ES	2	0
Paulo Lima	plima@keep.pt	ent_KEEP	7	0
Pedro Penteado	pedro.penteado@dglab.gov.pt	ent_DGLAB	5	1
Regina Neves Lopes	Regina.Lopes@sgmf.gov.pt	ent_SGMF	1	0
Ricardo Almeida	ricardo.almeida@dgeg.gov.pt	ent_DGEG	1	0
Ricardo Canela	tyty@tyty.pt	ent_BdP	3	0
Rui Araújo	pg38425@alunos.uminho.pt	ent_II	7	9
Rui Araújo Entidade	ruifilipearaujo@hotmail.com	ent_AAN	1	0
Rui Araújo Simples	0.rffa.0@gmail.com	ent_LNEC	1	0
Sandra Cristina Patrício da Silva	spatricio@mun-sines.pt	ent_CMSNS	1	0
Silvestre Lacerda	silvestre.lacerda@dglab.gov.pt	ent_DGLAB	5	0
Sónia Isabel Ferreira Gonçalves Negrão	sonia.negrao@cm-albufeira.pt	ent_CMABF	1	0
Sónia Patrícia Pinheiro Reis	sonia.reis@icnf.pt	ent_ICNF	1	0
Zélia Gomes	zelia.gomes@dglab.gov.pt	ent_DGLAB	4	0
clara cristina rainho viegas	clara.viegas@dglab.gov.pt	ent_DGLAB	4	0
jcm	jcm@live.com	ent_AAN	1	0
jcr-rep-entidade	zzglider@gmail.com	ent_A3ES	1	0
octavio	octaviojmaia@gmail.com	ent_A3ES	2	0
Élia Cristina Viegas Pedro	epedro@ccdr-alg.pt	ent_CCDR-Alg	1	0

Figura 14: Tabela exercício 6

7.1 Conclusão final e trabalho futuro

Através do desenvolvimento deste projecto conseguimos melhorar as nossas competências relativamente ao módulo *re* do *python* e no *html*. O módulo *re* do *python* provou ser altamente eficiente na leitura do ficheiro de texto em questão. Em suma, achamos que o nosso trabalho satisfaz os requisitos pedidos inicialmente no enunciado. Também achamos que as nossas competências em *python* e *html* saíram reforçadas após a concretização deste trabalho prático.