

Universidade do Minho
Mestrado Integrado em Engenharia Informática
IPLN-Trabalho Prático 1

João Bernardo Freitas A74814

André Vieira A78322

João Mendes A71862

13 de Novembro 2020

Contents

1	Introdução	3
2	Resolução	3
2.1	Adaptação	3
2.2	Limpeza	4
2.3	Árvore genealógica	4
3	Resultados	6
4	Conclusão	6

1 Introdução

Neste relatório iremos expor o processo de desenvolvimento do trabalho prático nº1 da Unidade Curricular de Introdução ao Processamento de Linguagem Natural. Este trabalho teve como objectivo a criação de uma árvore genealógica interactiva através dos ficheiro fornecidos na Aula 3.

2 Resolução

A nossa resolução consistiu em 3 fases distintas:

- Adaptação do trabalho feito na Aula 3 para resolver este problema
- Limpeza dos nomes de todos os indivíduos através de expressões regulares
- Criação de uma árvore genealógica interactiva através do jinja2 e páginas **html**

2.1 Adaptação

Na aula 3 a turma desenvolveu a função **gTree** que ao ser fornecida um código de um indivíduo iria dar o nome deste mesmo bem como o nome de todos os seus descendentes. Para a criação de uma árvore genealógica com páginas **html** esta solução não iria funcionar, como tal decidimos adapta-la para as nossas necessidades.

```
def gTree(n):
    nome = BG[n]['name']
    filho = []
    for i in BG[n]['fams']:
        for f in BF[i]['child']:
            filho.append((limpa(BG[f]['name']), f))
    return (limpa(nome), filho)
```

Figure 1: gTree adaptada

Essencialmente, esta função irá entregar um **tuplo** contendo o nome do indivíduo e uma lista de **tuplos** com o nome e código de cada filho.

*Nota: Na nossa solução o código de cada indivíduo é utilizado como nome da página, ou seja, o indivíduo com o código **I1** irá ter a sua própria página com o nome **I1.html**.*

2.2 Limpeza

Infelizmente o ficheiro que é utilizado como *input* tem muitos artefactos, por exemplo o nome da pessoa com código **I1**, *Victoria Hanover*, aparece desta forma *Victoria /Hanover/*. Como tal decidimos criar a função **limpa** que irá limpar todos os nomes desses artefactos.

```
#Funcao que remove todas os // e _ dos nomes
def limpa(t):
    t=re.sub(r"\_", " ",t)      #remove os _
    t=re.sub(r"\/+","",t)      #remove os / dos nomes
    return t
```

Figure 2: Função de limpeza

2.3 Árvore genealógica

Através do trabalho realizado nas duas fases anteriores podemos agora criar a árvore genealógica com assistência de templates do **jinja2**.

```
indice = []
for key, value in BG.items():
    filename = os.path.join(root, 'paginas', str(key) + ".html")
    indice.append([(limpa(BG[key]['name']), key)])
    f = gTree(key)
    nomes=f[1]
    #Adiciona um link para regressar ao inicio a todos os individuos
    nomes.append(("Voltar ao inicio","Index"))
    with open(filename, 'w') as fh:
        fh.write(template.render(
            h1=f[0],
            names=nomes
        ))
# Indice
filename = os.path.join(root, 'paginas', "Index.html")
with open(filename, 'w') as fh:
    fh.write(template2.render(
        h1="Indice",
        names=indice
    ))
```

Figure 3: Função que cria as páginas

Nesta função, o dicionário **BG** que foi desenvolvido na Aula 3 é percorrido de forma a obter os filhos, e seus códigos, de cada indivíduo, sendo que, através do **jinja2** e a informação obtida através do **gTree** podemos criar páginas **html** para cada indivíduo.

```
<h1>{{ h1 }}</h1>

<ul>
{% for name in names %}
  <li> <a href={{ name[1] }}.html>{{ name[0] }} </a></li>
{% endfor %}
</ul>
```

Figure 4: Template para as páginas

Finalmente é criado uma página **Index.html** que, tal como o nome indica, irá conter os nomes de todos os indivíduos bem como ligações para as respectivas páginas **html**.

Este índice é criado através de uma lista de tuplos contendo o nome e respetivo código de cada indivíduo e um novo *template* do **jinja2**.

```
<h1>{{ h1 }}</h1>

<ul>
{% for name in names %}
  <li> <a href={{ name[0][1] }}.html>{{ name[0][0] }} </a></li>
{% endfor %}
</ul>
```

Figure 5: Template para o índice

3 Resultados

Indice

- [Victoria Hanover](#)
- [Albert Augustus Charles](#)
- [Victoria Adelaide Mary](#)
- [Edward VII Wettin](#)
- [Alice Maud Mary](#)
- [Alfred Ernest Albert](#)
- [Helena Augusta Victoria](#)
- [Louise Caroline Alberta](#)
- [Arthur William Patrick](#)
- [Leopold George Duncan](#)
- [Beatrice Mary Victoria](#)
- [Alexandra of Denmark "Alix"](#)
- [Albert Victor Christian](#)
- [George V Windsor](#)
- [Louise Victoria Alexandra](#)
- [Victoria Alexandra Olga](#)
- [Maude Charlotte Mary](#)
- [John Alexander](#)
- [George Victor of Waldeck](#)
- [Frederick III](#)
- [William II](#)
- [Louis IV of Hesse](#)
- [Helena Frederica of Waldeck](#)
- [Alice of Athlone](#)
- [Henry Maurice of Battenberg](#)
- [Alexander of Carisbrooke](#)
- [Victoria Eugenie "Ena"](#)

Figure 6: Índice

Alfred Ernest Albert

- [Alfred](#)
- [Marie of Saxe-Coburg and Gotha](#)
- [Victoria Melita of Edinburgh](#)
- [Alexandra](#)
- [Beatrice](#)
- [Voltar ao inicio](#)

Figure 7: Página de um indivíduo

4 Conclusão

Através deste trabalho prático o grupo conseguiu aprofundar os conhecimentos sobre a linguagem **Python**, expressões regulares, **html** e **jinja2**.

Em suma, o grupo está satisfeito com o trabalho realizado pois sente-se capaz de conseguir utilizar as ferramentas no futuro se for necessário.