Programação II
+
Estruturas de Dados para Bioinformática

Formatos de dados

Hugo Pacheco

DCC/FCUP

22/23

Dados

- Nos ficheiros que temos visto até agora, os dados estão em texto livre
- Mas têm sempre alguma estrutura, por exemplo, hierarquia de cantos e estrofes nos lusíadas

“Canto Primeiro

1
As armas e os barões assinalados,
Que da ocidental praia Lusitana,
Por mares nunca de antes navegados,
Passaram ainda além da Taprobana,
Em perigos e guerras esforçados,
Mais do que prometia a força humana,
E entre gente remota edificaram
Novo Reino, que tanto sublimaram;
[…]”

Dados

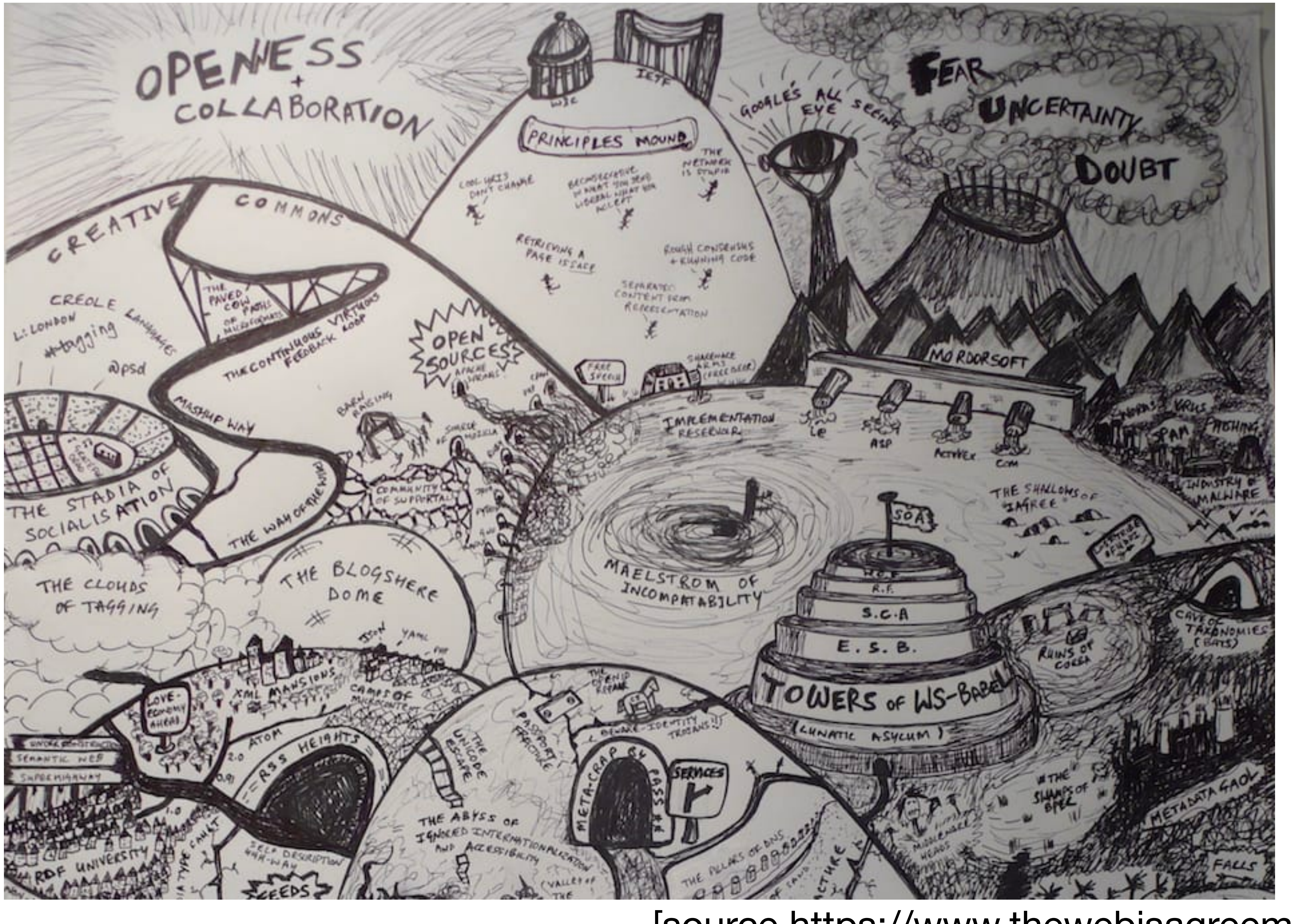
- Nos ficheiros que temos visto até agora, os dados estão em texto livre
- Mas têm sempre alguma estrutura, por exemplo, informação de enzimas por linha, separada por / *e* ‘

```
“REBASE version 302                                  staden.302

   =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
   REBASE, The Restriction Enzyme Database   http://rebase.neb.com
   Copyright (c)  Dr. Richard J. Roberts, 2023.   All rights reserved.
   =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

Rich Roberts                                    Jan 27 2023

AanI/TTA'TAA//
AarI/CACCTGCNNNN'NNNN/'NNNNNNNNGCAGGTG//
AasI/GACNNNN'NNGTC//
AatII/GACGT'C//
[…]”
```

Dados

- Para ler os dados temos que explorar a sua estrutura.
- Mas se estão em texto livre…
  - pode ser difícil encontrar padrões
  - cada caso é um caso
- Idealmente, diferentes conjuntos de dados devem seguir uma estrutura comum:
  - facilitar a partilha
  - garantir preservação e suporte de longo prazo
  - melhores ferramentas

“The Web is Agreement”



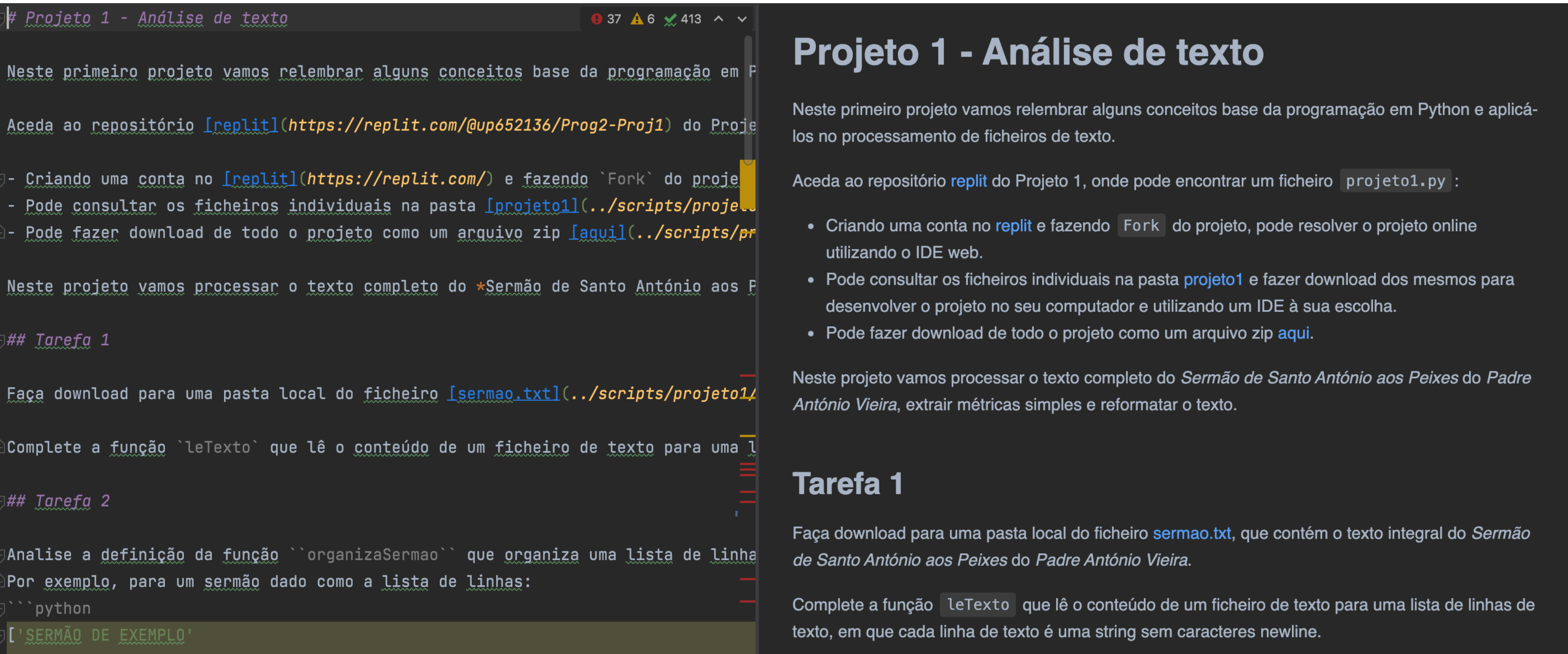
[source https://www.thewebisagreement.com/]

Formatos de dados

- Alguns dos formatos de dados mais comuns:
  - **Markdown (MD)**
  - HyperText Markup Language (HTML)
  - **Comma Separated Value (CSV)**
  - Excel (XLS)
  - **JavaScript Object Notation (JSON)**
  - eXtended Markup Language (XML)
  - Structured Query Language (SQL)
  - …
- Fáceis de ler em Python!

Markdown

- ficheiros de texto em que se usa caracteres especiais para formatação
- formato “standard” para notas, blogs ou gerar HTML



~~~
# Projeto 1 - Análise de texto

Neste primeiro projeto vamos relembrar alguns conceitos base da programação em P

Aceda ao repositório [replit](https://replit.com/@up652136/Prog2-Proj1) do Proje

- Criando uma conta no [replit](https://replit.com/) e fazendo `Fork` do proje
- Pode consultar os ficheiros individuais na pasta [projeto1](../scripts/proje
- Pode fazer download de todo o projeto como um arquivo zip [aqui](../scripts/pr

Neste projeto vamos processar o texto completo do *Sermão de Santo António aos P

## Tarefa 1

Faça download para uma pasta local do ficheiro [sermao.txt](../scripts/projeto1/

Complete a função `leTexto` que lê o conteúdo de um ficheiro de texto para uma l

## Tarefa 2

Analise a definição da função ``organizaSermao`` que organiza uma lista de linha
Por exemplo, para um sermão dado como a lista de linhas:
```python
['SERMÃO DE EXEMPLO'
~~~

Projeto 1 - Análise de texto

Neste primeiro projeto vamos relembrar alguns conceitos base da programação em Python e aplicá-los no processamento de ficheiros de texto.

Aceda ao repositório replit do Projeto 1, onde pode encontrar um ficheiro `projeto1.py`:

- Criando uma conta no replit e fazendo `Fork` do projeto, pode resolver o projeto online utilizando o IDE web.
- Pode consultar os ficheiros individuais na pasta projeto1 e fazer download dos mesmos para desenvolver o projeto no seu computador e utilizando um IDE à sua escolha.
- Pode fazer download de todo o projeto como um arquivo zip aqui.

Neste projeto vamos processar o texto completo do *Sermão de Santo António aos Peixes* do *Padre António Vieira*, extrair métricas simples e reformatar o texto.

Tarefa 1

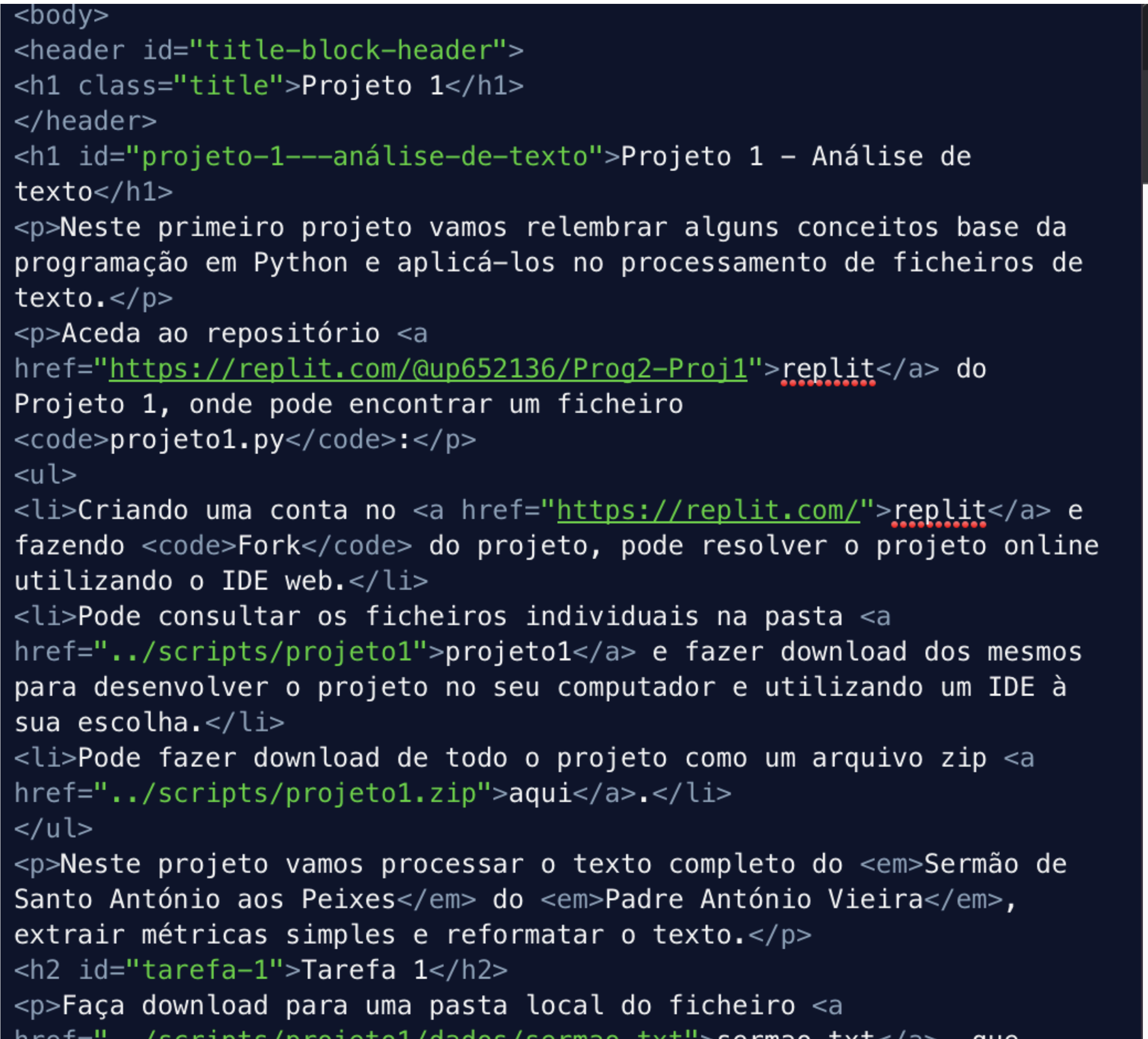
Faça download para uma pasta local do ficheiro sermao.txt, que contém o texto integral do *Sermão de Santo António aos Peixes* do *Padre António Vieira*.

Complete a função `leTexto` que lê o conteúdo de um ficheiro de texto para uma lista de linhas de texto, em que cada linha de texto é uma string sem caracteres newline.

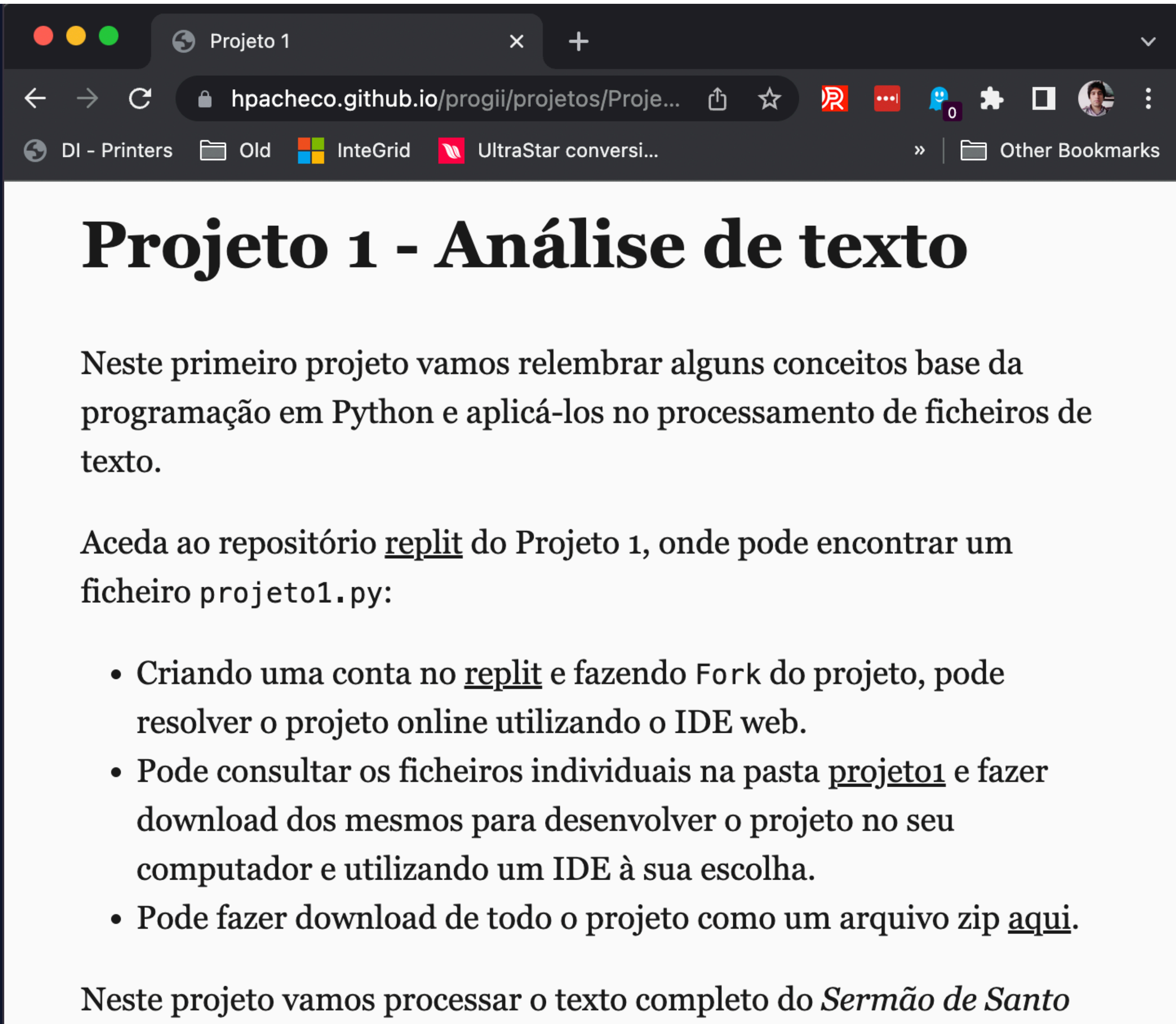
HTML

- ficheiros de texto em que se usa a notação ***<campo>***_texto_***<campo/>*** para definir uma estrutura hierárquica

- formato “standard” para representar páginas web



```
<body>
<header id="title-block-header">
<h1 class="title">Projeto 1</h1>
</header>
<h1 id="projeto-1---análise-de-texto">Projeto 1 - Análise de
texto</h1>
<p>Neste primeiro projeto vamos relembrar alguns conceitos base da
programação em Python e aplicá-los no processamento de ficheiros de
texto.</p>
<p>Aceda ao repositório <a
href="https://replit.com/@up652136/Prog2-Proj1">replit</a> do
Projeto 1, onde pode encontrar um ficheiro
<code>projeto1.py</code>:</p>
<ul>
<li>Criando uma conta no <a href="https://replit.com/">replit</a> e
fazendo <code>Fork</code> do projeto, pode resolver o projeto online
utilizando o IDE web.</li>
<li>Pode consultar os ficheiros individuais na pasta <a
href="../scripts/projeto1">projeto1</a> e fazer download dos mesmos
para desenvolver o projeto no seu computador e utilizando um IDE à
sua escolha.</li>
<li>Pode fazer download de todo o projeto como um arquivo zip <a
href="../scripts/projeto1.zip">aqui</a>.</li>
</ul>
<p>Neste projeto vamos processar o texto completo do <em>Sermão de
Santo António aos Peixes</em> do <em>Padre António Vieira</em>,
extrair métricas simples e reformatar o texto.</p>
<h2 id="tarefa-1">Tarefa 1</h2>
<p>Faça download para uma pasta local do ficheiro <a
href="../scripts/projeto1/dados/sermao.txt">sermao.txt</a>, que
```



Projeto 1 - Análise de texto

Neste primeiro projeto vamos relembrar alguns conceitos base da programação em Python e aplicá-los no processamento de ficheiros de texto.

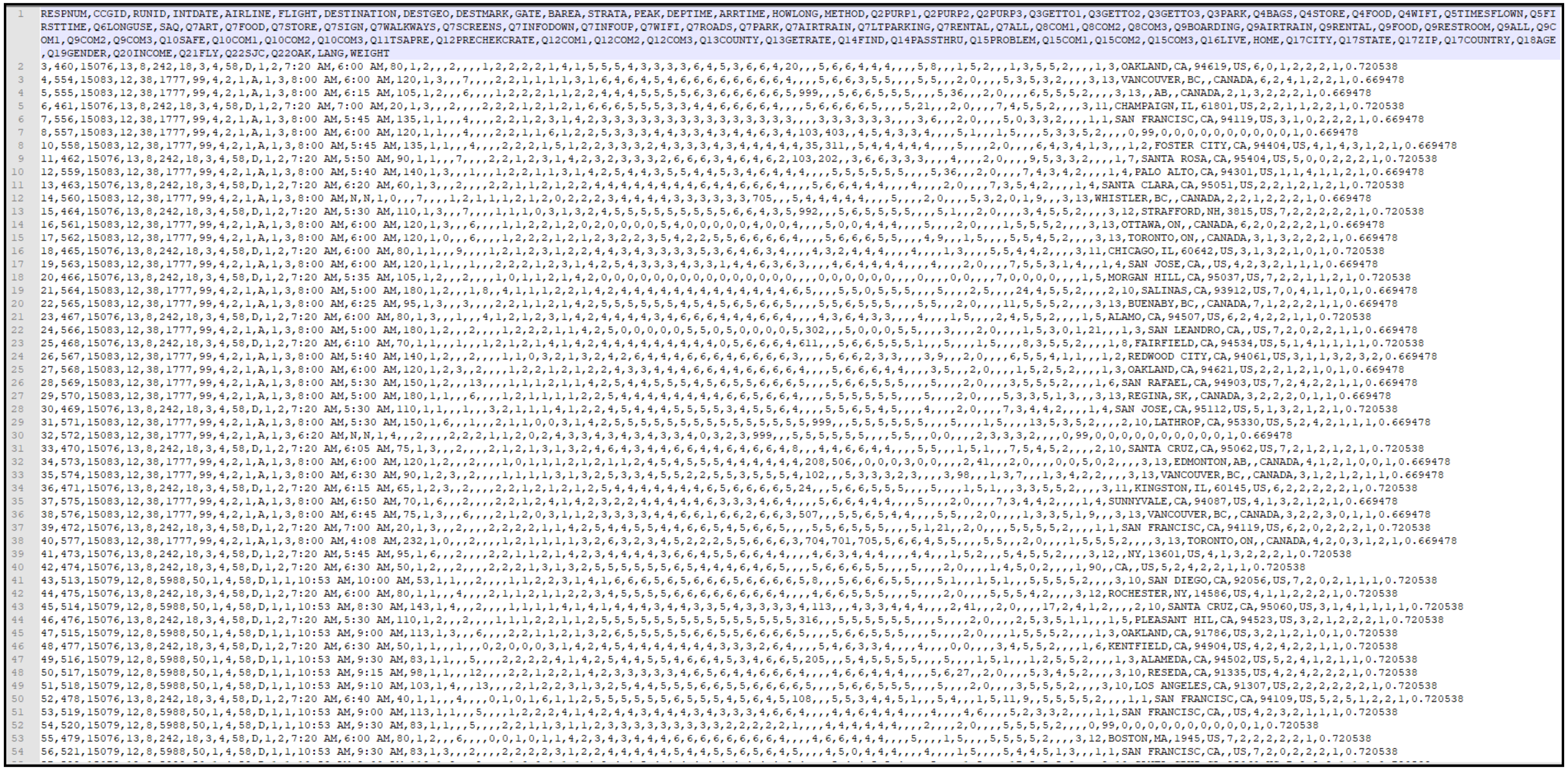
Aceda ao repositório replit do Projeto 1, onde pode encontrar um ficheiro `projeto1.py`:

- Criando uma conta no replit e fazendo `Fork` do projeto, pode resolver o projeto online utilizando o IDE web.
- Pode consultar os ficheiros individuais na pasta projeto1 e fazer download dos mesmos para desenvolver o projeto no seu computador e utilizando um IDE à sua escolha.
- Pode fazer download de todo o projeto como um arquivo zip aqui.

Neste projeto vamos processar o texto completo do *Sermão de Santo*

CSV

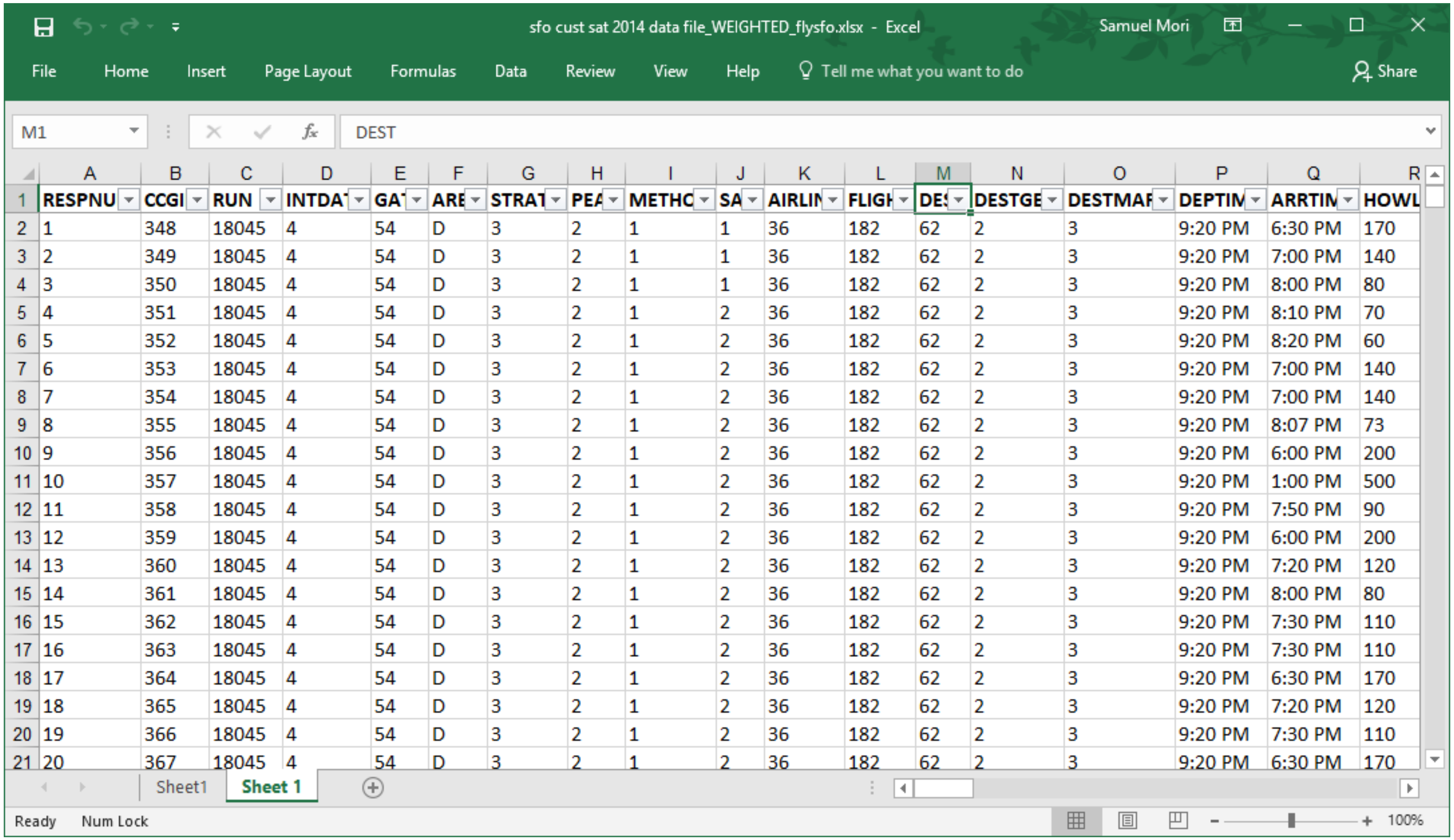
- ficheiros de texto em que se usa vírgula para separar valores
- formato “standard” para comunicar dados tabulares



```
RESPNUM,CCGID,RUNID,INTDATE,AIRLINE,FLIGHT,DESTINATION,DESTGEO,DESTMARK,GATE,BAREA,STRATA,PEAK,DEPTIME,ARRTIME,HOWLONG,METHOD,Q2PURP1,Q2PURP2,Q2PURP3,Q3GETTO1,Q3GETTO2,Q3GETTO3,Q3PARK,Q4BAGS,Q4STORE,Q4FOOD,Q4WIFI,Q5TIMESFLOWN,Q5FIRSTTIME,Q6LONGUSE,SAQ,Q7ART,Q7FOOD,Q7STORE,Q7SIGN,Q7WALKWAYS,Q7SCREENS,Q7INFODOWN,Q7INFOUP,Q7WIFI,Q7ROADS,Q7PARK,Q7AIRTRAIN,Q7LTPARKING,Q7RENTAL,Q7ALL,Q8COM1,Q8COM2,Q8COM3,Q9BOARDING,Q9AIRTRAIN,Q9RENTAL,Q9FOOD,Q9RESTROOM,Q9ALL,Q9COM1,Q9COM2,Q9COM3,Q10SAFE,Q10COM1,Q10COM2,Q10COM3,Q11TSAPRE,Q12PRECHEKCRATE,Q12COM1,Q12COM2,Q12COM3,Q13COUNTY,Q13GETRATE,Q14FIND,Q14PASSTHRU,Q15PROBLEM,Q15COM1,Q15COM2,Q15COM3,Q16LIVE,HOME,Q17CITY,Q17STATE,Q17ZIP,Q17COUNTRY,Q18AGE,Q19GENDER,Q20INCOME,Q21FLY,Q22SJC,Q22OAK,LANG,WEIGHT
3,460,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:00 AM,80,1,2,,,2,,,,1,2,2,2,2,1,4,1,5,5,5,4,3,3,3,3,6,4,5,3,6,6,4,20,,,5,6,6,4,4,4,,,,5,8,,,1,5,2,,,1,3,5,5,2,,,,1,3,OAKLAND,CA,94619,US,6,0,1,2,2,2,1,0.720538
4,554,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:00 AM,120,1,3,,,7,,,,2,2,1,1,1,1,3,1,6,4,6,4,5,4,6,6,6,6,6,6,6,4,,,,5,6,6,3,5,5,,,,5,5,,,2,0,,,,5,3,5,3,2,,,,3,13,VANCOUVER,BC,,CANADA,6,2,4,1,2,2,1,0.669478
5,555,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:15 AM,105,1,2,,,6,,,,1,2,2,2,1,1,2,2,4,4,4,5,5,5,5,6,3,6,6,6,6,6,5,999,,,5,6,6,5,5,5,,,,5,36,,,2,0,,,,6,5,5,5,2,,,,3,13,,AB,,CANADA,2,1,3,2,2,2,1,0.669478
6,461,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,7:00 AM,20,1,3,,,2,,,,2,2,2,1,2,1,2,1,6,6,6,5,5,5,3,3,4,4,6,6,6,6,4,,,,5,6,6,6,6,5,,,,5,21,,,2,0,,,,7,4,5,5,2,,,,3,11,CHAMPAIGN,IL,61801,US,2,2,1,1,2,2,1,0.720538
7,556,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,5:45 AM,135,1,1,,,4,,,,2,2,1,2,3,1,4,2,3,3,3,3,3,3,3,3,3,3,3,3,3,3,,,,3,3,3,3,3,3,,,,3,6,,,2,0,,,,5,0,3,3,2,,,,1,1,SAN FRANCISC,CA,94119,US,3,1,0,2,2,2,1,0.669478
8,557,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:00 AM,120,1,1,,,4,,,,2,2,1,1,6,1,2,2,5,3,3,3,4,4,3,3,4,3,4,4,6,3,4,103,403,,4,5,4,3,3,4,,,,5,1,,,1,5,,,,5,3,3,5,2,,,,0,99,0,0,0,0,0,0,0,0,0,1,0.669478
10,558,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,5:45 AM,135,1,1,,,4,,,,2,2,2,1,5,1,2,2,3,3,3,2,4,3,3,3,4,3,4,4,4,4,4,35,311,,5,4,4,4,4,4,,,,5,,,,2,0,,,,6,4,3,4,1,3,,,1,2,FOSTER CITY,CA,94404,US,4,1,4,3,1,2,1,0.669478
11,462,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,5:50 AM,90,1,1,,,7,,,,2,2,1,2,3,1,4,2,3,2,3,3,3,2,6,6,6,3,4,6,4,6,2,103,202,,3,6,6,3,3,3,,,,4,,,,2,0,,,,9,5,3,3,2,,,,1,7,SANTA ROSA,CA,95404,US,5,0,0,2,2,2,1,0.720538
12,559,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,5:40 AM,140,1,3,,,1,,,,1,2,2,1,1,3,1,4,2,5,4,4,3,5,5,4,4,5,3,4,6,4,4,4,,,,5,5,5,5,5,5,,,,5,36,,,2,0,,,,7,4,3,4,2,,,,1,4,PALO ALTO,CA,94301,US,1,1,4,1,1,2,1,0.669478
13,463,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:20 AM,60,1,3,,,2,,,,2,2,1,1,2,1,2,2,4,4,4,4,4,4,4,4,6,4,4,6,6,6,4,,,,5,6,6,4,4,4,,,,4,,,,2,0,,,,7,3,5,4,2,,,,1,4,SANTA CLARA,CA,95051,US,2,2,1,2,1,2,1,0.720538
14,560,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,N,N,1,0,,,7,,,,1,2,1,1,1,2,1,2,0,2,2,2,3,4,4,4,4,3,3,3,3,3,3,705,,,5,4,4,4,4,4,,,,5,,,,2,0,,,,5,3,2,0,1,9,,,3,13,WHISTLER,BC,,CANADA,2,2,1,2,2,2,1,0.669478
15,464,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,5:30 AM,110,1,3,,,7,,,,1,1,1,0,3,1,3,2,4,5,5,5,5,5,5,5,5,5,6,6,4,3,5,992,,,5,6,5,5,5,5,,,,5,1,,,2,0,,,,3,4,5,5,2,,,,3,12,STRAFFORD,NH,3815,US,7,2,2,2,2,2,1,0.720538
16,561,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:00 AM,120,1,3,,,6,,,,1,1,2,2,1,2,0,2,0,0,0,0,5,4,0,0,0,0,0,4,0,0,4,,,,5,0,0,4,4,4,,,,5,,,,2,0,,,,1,5,5,5,2,,,,3,13,OTTAWA,ON,,CANADA,6,2,0,2,2,2,1,0.669478
17,562,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:00 AM,120,1,0,,,6,,,,1,2,2,2,1,2,1,2,3,2,2,3,5,4,2,2,5,5,6,6,6,6,4,,,,5,6,6,6,5,5,,,,4,9,,,1,5,,,,5,5,4,5,2,,,,3,13,TORONTO,ON,,CANADA,3,1,3,2,2,2,1,0.669478
18,465,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:00 AM,80,1,1,,,9,,,,1,2,1,2,3,1,2,2,4,4,3,4,3,3,3,3,5,3,6,4,6,3,4,,,,4,3,2,4,4,4,,,,4,,,,1,3,,,,5,5,4,4,2,,,,3,11,CHICAGO,IL,60642,US,3,1,3,2,1,0,1,0.720538
19,563,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:00 AM,120,1,1,,,1,,,2,2,2,1,2,3,1,4,2,5,4,3,3,3,4,3,3,1,4,4,6,3,6,3,,,,4,6,4,4,4,4,,,,4,,,,2,0,,,,7,5,5,3,1,4,,,1,4,SAN JOSE,CA,,US,4,2,3,2,1,1,1,0.669478
20,466,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,5:35 AM,105,1,2,,,2,,,,1,0,1,1,2,1,4,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,,,,0,0,0,0,0,0,,,,0,,,,0,0,,,,7,0,0,0,0,,,,1,5,MORGAN HILL,CA,95037,US,7,2,2,1,1,2,1,0.720538
21,564,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,5:00 AM,180,1,2,,,1,8,,4,1,1,1,2,2,1,4,2,4,4,4,4,4,4,4,4,4,4,4,4,4,6,5,,,,5,5,0,5,5,5,,,,5,,,,2,5,,,,24,4,5,5,2,,,,2,10,SALINAS,CA,93912,US,7,0,4,1,1,0,1,0.669478
22,565,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:25 AM,95,1,3,,,3,,,,2,2,1,1,2,1,4,2,5,5,5,5,5,5,4,5,4,5,6,5,6,6,5,,,,5,5,6,5,5,5,,,,5,5,,,2,0,,,,11,5,5,5,2,,,,3,13,BUENABY,BC,,CANADA,7,1,2,2,2,1,1,0.669478
23,467,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:00 AM,80,1,3,,,1,,,4,1,2,1,2,3,1,4,2,4,4,4,4,3,4,6,6,6,4,4,4,6,6,4,,,,4,3,6,4,3,3,,,,4,,,,1,5,,,,2,4,5,5,2,,,,1,5,ALAMO,CA,94507,US,6,2,4,2,2,1,1,0.720538
24,566,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,5:00 AM,180,1,2,,,2,,,,1,2,2,2,1,1,4,2,5,0,0,0,0,0,5,5,0,5,0,0,0,0,5,302,,,5,0,0,0,5,5,,,,3,,,,2,0,,,,1,5,3,0,1,21,,,1,3,SAN LEANDRO,CA,,US,7,2,0,2,2,1,1,0.669478
25,468,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:10 AM,70,1,1,,,1,,,1,2,1,2,1,4,1,4,2,4,4,4,4,4,4,4,4,4,0,5,6,6,6,4,611,,,5,6,6,5,5,5,1,,,5,,,,1,5,,,,8,3,5,5,2,,,,1,8,FAIRFIELD,CA,94534,US,5,1,4,1,1,1,1,0.720538
26,567,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,5:40 AM,140,1,2,,,2,,,,1,1,0,3,2,1,3,2,4,2,6,4,4,4,6,6,6,4,6,6,6,6,3,,,,5,6,6,2,3,3,,,,3,9,,,2,0,,,,6,5,5,4,1,1,,,1,2,REDWOOD CITY,CA,94061,US,3,1,1,3,2,3,2,0.669478
27,568,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:00 AM,120,1,2,3,,2,,,,1,2,2,1,2,1,2,2,4,3,3,4,4,4,6,6,4,4,6,6,6,6,4,,,,5,6,6,6,4,4,,,,3,5,,,2,0,,,,1,5,2,5,2,,,,1,3,OAKLAND,CA,94621,US,2,2,1,2,1,0,1,0.669478
28,569,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,5:30 AM,150,1,2,,,13,,,,1,1,1,2,1,1,4,2,5,4,4,5,5,5,4,5,6,5,5,6,6,6,5,,,,5,6,6,5,5,5,,,,5,,,,2,0,,,,3,5,5,5,2,,,,1,6,SAN RAFAEL,CA,94903,US,7,2,4,2,2,1,1,0.669478
29,570,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,5:00 AM,180,1,1,,,6,,,,1,2,1,1,1,1,2,2,5,4,4,4,4,4,4,4,4,6,6,5,6,6,4,,,,5,5,5,5,5,5,,,,5,,,,2,0,,,,5,3,3,5,1,3,,,3,13,REGINA,SK,,CANADA,3,2,2,2,0,1,1,0.669478
30,469,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,5:30 AM,110,1,1,,,1,,,3,2,1,1,1,4,1,2,2,4,5,4,4,4,5,5,5,5,3,4,5,5,6,4,,,,5,5,6,5,4,5,,,,4,,,,2,0,,,,7,3,4,4,2,,,,1,4,SAN JOSE,CA,95112,US,5,1,3,2,1,2,1,0.720538
31,571,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,5:30 AM,150,1,6,,,1,,,2,1,1,0,0,3,1,4,2,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,999,,,5,5,5,5,5,5,,,,5,,,,1,5,,,,13,5,3,5,2,,,,2,10,LATHROP,CA,95330,US,5,2,4,2,1,1,1,0.669478
32,572,15083,12,38,1777,99,4,2,1,A,1,3,6:20 AM,N,N,1,4,,,2,,,,2,2,2,1,1,2,0,2,4,3,3,4,3,4,3,4,3,3,4,0,3,2,3,999,,,5,5,5,5,5,5,,,,5,5,,,0,0,,,,2,3,3,3,2,,,,0,99,0,0,0,0,0,0,0,0,0,0,1,0.669478
33,470,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:05 AM,75,1,3,,,2,,,,2,1,2,1,3,1,3,2,4,6,4,3,4,4,6,6,4,4,6,4,6,6,4,8,,,4,4,6,6,4,4,,,,5,5,,,1,5,1,,,7,5,4,5,2,,,,2,10,SANTA CRUZ,CA,95062,US,7,2,1,2,1,2,1,0.720538
34,573,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:00 AM,120,1,2,,,2,,,,1,0,1,1,1,2,1,2,1,1,2,4,5,4,5,5,5,4,4,4,4,4,4,208,506,,0,0,0,3,0,0,,,,2,41,,,2,0,,,,0,0,5,0,2,,,,3,13,EDMONTON,AB,,CANADA,4,1,2,1,0,0,1,0.669478
35,574,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:30 AM,90,1,2,3,,2,,,,1,1,1,1,3,1,3,2,5,3,3,4,5,5,2,2,5,5,3,5,5,5,4,102,,,5,3,3,3,2,3,,,,3,98,,,1,3,7,,,1,3,4,2,2,,,,3,13,VANCOUVER,BC,,CANADA,3,1,2,1,2,1,1,0.669478
36,471,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:15 AM,65,1,2,3,,2,,,,2,2,1,2,1,2,1,2,5,4,4,4,4,4,4,6,5,6,6,6,6,5,24,,,5,6,6,5,5,5,,,,5,,,,1,5,1,,,3,3,5,5,2,,,,3,11,KINGSTON,IL,60145,US,6,2,2,2,2,2,1,0.720538
37,575,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:50 AM,70,1,6,,,2,,,,2,2,1,2,4,1,4,2,3,2,2,4,4,4,4,4,6,3,3,3,4,6,4,,,,5,6,6,4,4,4,,,,5,,,,2,0,,,,7,3,4,4,2,,,,1,4,SUNNYVALE,CA,94087,US,4,1,3,2,1,2,1,0.669478
38,576,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,6:45 AM,75,1,3,,,6,,,,2,1,2,0,3,1,1,2,3,3,3,3,4,4,6,6,1,6,6,2,6,6,3,507,,,5,5,6,5,4,4,,,,5,5,,,2,0,,,,1,3,3,5,1,9,,,3,13,VANCOUVER,BC,,CANADA,3,2,2,3,0,1,1,0.669478
39,472,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,7:00 AM,20,1,3,,,2,,,,2,2,2,2,1,1,4,2,5,4,4,5,5,4,4,6,6,5,4,5,6,6,5,,,,5,5,6,5,5,5,,,,5,1,21,,2,0,,,,5,5,5,5,2,,,,1,1,SAN FRANCISC,CA,94119,US,6,2,0,2,2,2,1,0.720538
40,577,15083,12,38,1777,99,4,2,1,A,1,3,8:00 AM,4:08 AM,232,1,0,,,2,,,,1,2,1,1,1,1,3,2,6,3,2,3,4,5,2,2,2,5,5,6,6,6,3,704,701,705,5,6,6,4,5,5,,,,5,5,,,2,0,,,,1,5,5,5,2,,,,3,13,TORONTO,ON,,CANADA,4,2,0,3,1,2,1,0.669478
41,473,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,5:45 AM,95,1,6,,,2,,,,2,2,1,1,2,1,4,2,3,4,4,4,4,3,6,6,4,5,5,6,6,4,4,,,,4,6,3,4,4,4,,,,4,4,,,1,5,2,,,5,4,5,5,2,,,,3,12,,NY,13601,US,4,1,3,2,2,2,1,0.720538
42,474,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:30 AM,50,1,2,,,2,,,,2,2,2,1,3,1,3,2,5,5,5,5,5,5,6,5,4,4,4,6,4,6,5,,,,5,6,6,6,5,5,,,,5,,,,2,0,,,,1,4,5,0,2,,,,1,90,,CA,,US,5,2,4,2,2,1,1,0.720538
43,513,15079,12,8,5988,50,1,4,58,D,1,1,10:53 AM,10:00 AM,53,1,1,,,2,,,,1,1,2,2,3,1,4,1,6,6,6,5,6,5,6,6,6,5,6,6,6,6,5,8,,,5,6,6,6,5,5,,,,5,1,,,1,5,1,,,5,5,5,5,2,,,,3,10,SAN DIEGO,CA,92056,US,7,2,0,2,1,1,1,0.720538
44,475,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:00 AM,80,1,1,,,4,,,,2,1,1,2,1,1,2,2,3,4,5,5,5,5,6,6,6,6,6,6,6,6,4,,,,4,6,6,5,5,5,,,,5,,,,2,0,,,,5,5,5,4,2,,,,3,12,ROCHESTER,NY,14586,US,4,1,1,2,2,2,1,0.720538
45,514,15079,12,8,5988,50,1,4,58,D,1,1,10:53 AM,8:30 AM,143,1,4,,,2,,,,1,1,1,1,4,1,4,1,4,4,4,3,4,4,3,3,5,4,3,3,3,3,4,113,,,4,3,3,4,4,4,,,,2,41,,,2,0,,,,17,2,4,1,2,,,,2,10,SANTA CRUZ,CA,95060,US,3,1,4,1,1,1,1,0.720538
46,476,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,5:30 AM,110,1,2,,,2,,,,1,1,1,2,2,1,1,2,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,316,,,5,5,5,5,5,5,,,,5,,,,2,0,,,,2,5,3,5,1,1,,,1,5,PLEASANT HIL,CA,94523,US,3,2,1,2,2,2,1,0.720538
47,515,15079,12,8,5988,50,1,4,58,D,1,1,10:53 AM,9:00 AM,113,1,3,,,6,,,,2,2,1,1,2,1,3,2,6,5,5,5,5,5,6,6,5,5,6,6,6,6,5,,,,5,6,6,5,5,5,,,,5,,,,2,0,,,,1,5,5,5,2,,,,1,3,OAKLAND,CA,91786,US,3,2,1,2,1,0,1,0.720538
48,477,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:30 AM,50,1,1,,,1,,,0,2,0,0,0,3,1,4,2,4,5,4,4,4,4,4,4,4,3,3,3,2,6,4,,,,5,4,6,3,3,4,,,,4,,,,0,0,,,,3,4,5,5,2,,,,1,6,KENTFIELD,CA,94904,US,4,2,4,2,2,1,1,0.720538
49,516,15079,12,8,5988,50,1,4,58,D,1,1,10:53 AM,9:30 AM,83,1,1,,,5,,,,2,2,2,2,4,1,4,2,5,4,4,5,5,4,6,6,4,5,3,4,6,6,5,205,,,5,4,5,5,5,5,,,,5,,,,1,5,1,,,1,2,5,5,2,,,,1,3,ALAMEDA,CA,94502,US,5,2,4,1,2,1,1,0.720538
50,517,15079,12,8,5988,50,1,4,58,D,1,1,10:53 AM,9:15 AM,98,1,1,,,12,,,,2,2,1,2,2,1,4,2,3,3,3,3,3,4,6,5,6,4,4,6,6,6,4,,,,4,6,6,4,4,4,,,,5,6,27,,2,0,,,,5,3,4,5,2,,,,3,10,RESEDA,CA,91335,US,4,2,4,2,2,2,1,0.720538
51,518,15079,12,8,5988,50,1,4,58,D,1,1,10:53 AM,9:10 AM,103,1,4,,,13,,,,2,1,2,2,3,1,3,2,5,4,4,5,5,5,6,6,5,5,6,6,6,6,5,,,,5,6,6,5,5,5,,,,5,,,,2,0,,,,3,5,5,5,2,,,,3,10,LOS ANGELES,CA,91307,US,2,2,2,2,2,2,1,0.720538
52,478,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:40 AM,40,1,1,,,4,,,,0,1,0,1,6,1,1,2,5,5,5,5,5,5,6,5,5,5,4,5,6,4,5,108,,,5,5,3,4,4,5,1,,,5,4,,,1,5,11,9,,,5,5,5,5,2,,,,1,1,SAN FRANCISC,CA,94109,US,5,2,5,1,2,2,1,0.720538
53,519,15079,12,8,5988,50,1,4,58,D,1,1,10:53 AM,9:00 AM,113,1,1,,,5,,,,1,2,2,2,4,1,4,2,4,4,3,4,4,4,3,4,3,3,3,4,6,6,4,,,,4,4,6,4,4,4,,,,4,,,,4,6,,,,5,2,3,3,2,,,,1,1,SAN FRANCISC,CA,,US,4,2,3,2,1,1,1,0.720538
54,520,15079,12,8,5988,50,1,4,58,D,1,1,10:53 AM,9:30 AM,83,1,1,,,5,,,,2,2,1,1,3,1,1,2,3,3,3,3,3,3,3,3,3,2,2,2,2,2,1,,,,4,4,4,4,4,4,,,,2,,,,2,0,,,,5,5,5,5,2,,,,0,99,0,0,0,0,0,0,0,0,0,0,1,0.720538
55,479,15076,13,8,242,18,3,4,58,D,1,2,7:20 AM,6:00 AM,80,1,2,,,6,,,,0,0,1,0,1,1,4,2,3,4,3,4,4,4,6,6,6,6,6,6,6,6,4,,,,4,6,6,4,4,4,,,,5,,,,1,5,,,,5,5,5,5,2,,,,3,12,BOSTON,MA,1945,US,7,2,2,2,2,2,1,0.720538
56,521,15079,12,8,5988,50,1,4,58,D,1,1,10:53 AM,9:30 AM,83,1,3,,,2,,,,2,2,2,2,3,1,2,2,4,4,4,4,4,5,4,4,5,5,6,5,6,4,5,,,,4,5,0,4,4,4,,,,4,,,,1,5,,,,5,4,4,5,1,3,,,1,1,SAN FRANCISC,CA,,US,7,2,0,2,2,2,1,0.720538
```

Excel

- O mesmo exemplo em folhas de cálculo



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	RESPNU	CCGI	RUN	INTDA	GA	ARE	STRAT	PEA	METHC	SA	AIRLIN	FLIGH	DES	DESTGE	DESTMAF	DEPTIN	ARRTIN	HOWL
2	1	348	18045	4	54	D	3	2	1	1	36	182	62	2	3	9:20 PM	6:30 PM	170
3	2	349	18045	4	54	D	3	2	1	1	36	182	62	2	3	9:20 PM	7:00 PM	140
4	3	350	18045	4	54	D	3	2	1	1	36	182	62	2	3	9:20 PM	8:00 PM	80
5	4	351	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	8:10 PM	70
6	5	352	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	8:20 PM	60
7	6	353	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	7:00 PM	140
8	7	354	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	7:00 PM	140
9	8	355	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	8:07 PM	73
10	9	356	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	6:00 PM	200
11	10	357	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	1:00 PM	500
12	11	358	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	7:50 PM	90
13	12	359	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	6:00 PM	200
14	13	360	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	7:20 PM	120
15	14	361	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	8:00 PM	80
16	15	362	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	7:30 PM	110
17	16	363	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	7:30 PM	110
18	17	364	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	6:30 PM	170
19	18	365	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	7:20 PM	120
20	19	366	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	7:30 PM	110
21	20	367	18045	4	54	D	3	2	1	2	36	182	62	2	3	9:20 PM	6:30 PM	170

CSV

- Ler um ficheiro CSV em Python
- E.g., índice de secura mensal para o Porto publicado pelo IPMA aqui

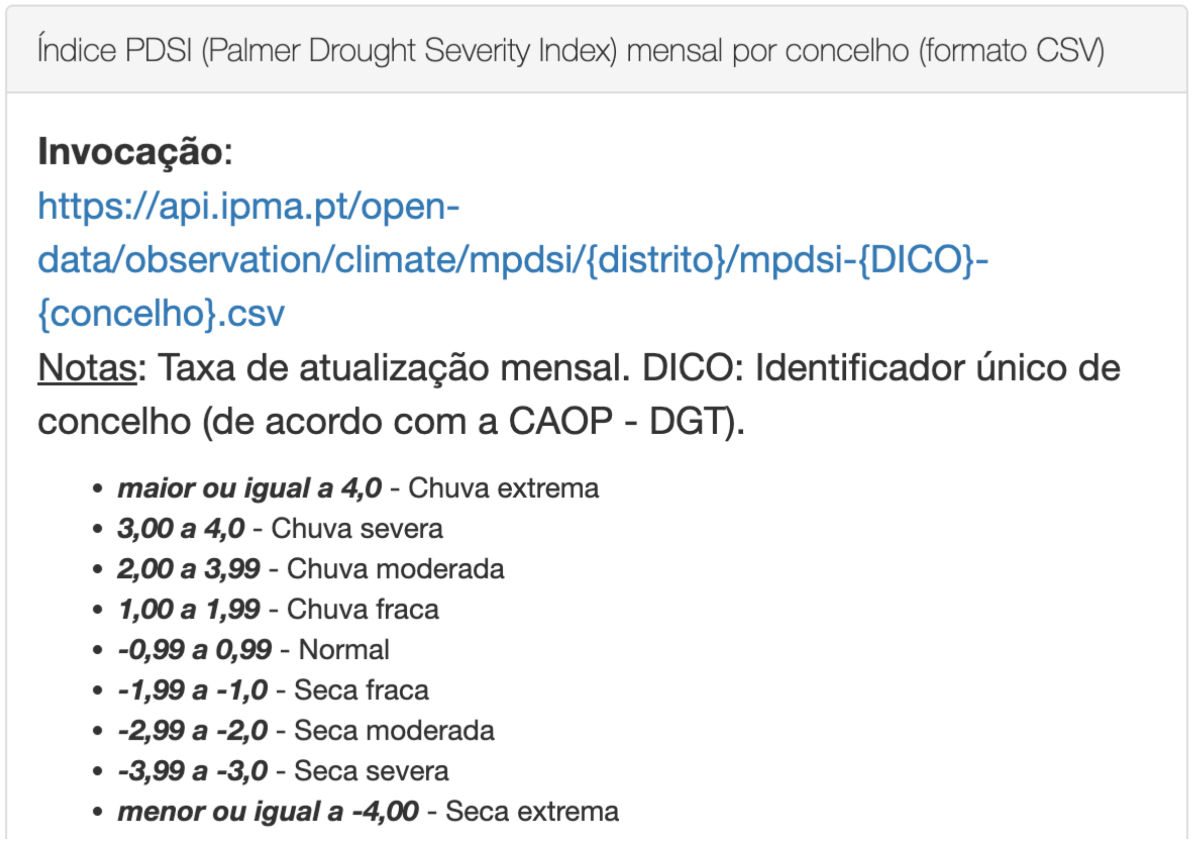
```
import csv
with open('../../dados/mpdsi-1312-porto.csv','r') as f:
    table = csv.reader(f)
    data = list(table)
print(data)
```

- Primeira linha é o cabeçalho que define o tipo de cada coluna
- Cada linha é uma lista de comprimento igual, com entradas de cada mês

```
['date', 'minimum', 'maximum', 'range', 'mean', 'std']
['2022-04-01', '-2.89150762558', '-2.75911641121', '0.132391214371', '-2.83662211895', '0.0348872640209']
['2022-05-01', '-3.8392894268', '-3.70847082138', '0.130818605423', '-3.7754253887', '0.0342999110149']
```

CSV

- Criar dicionário de classificações


Índice PDSI (Palmer Drought Severity Index) mensal por concelho (formato CSV)

Invocação:
https://api.ipma.pt/open-data/observation/climate/mpdsi/{distrito}/mpdsi-{DICO}-{concelho}.csv
Notas: Taxa de atualização mensal. DICO: Identificador único de concelho (de acordo com a CAOP - DGT).

- ***maior ou igual a 4,0*** - Chuva extrema
- ***3,00 a 4,0*** - Chuva severa
- ***2,00 a 3,99*** - Chuva moderada
- ***1,00 a 1,99*** - Chuva fraca
- ***-0,99 a 0,99*** - Normal
- ***-1,99 a -1,0*** - Seca fraca
- ***-2,99 a -2,0*** - Seca moderada
- ***-3,99 a -3,0*** - Seca severa
- ***menor ou igual a -4,00*** - Seca extrema

```
cabecalho=data[0]
meses = data[1:]

def classifica(n):
    if   n>=4: return 'chuva extrema'
    elif n>=3: return 'chuva severa'
    elif n>=2: return 'chuva moderada'
    elif n>=1: return 'chuva fraca'
    elif n>-1: return 'normal'
    elif n>-2: return 'seca fraca'
    elif n>-3: return 'seca moderada'
    elif n>-4: return 'seca severa'
    else     : return 'seca extrema’

meses_cs = { mes[0] : [classifica(float(n)) for n in mes[1:]]\
          for mes in meses }
```

CSV

- Selecionar a previsão mais recorrente por mês

```
def count(xs):
    c = {}
    for x in xs: c[x] = 1 + c.get(x,0)
    return c

def max_count(xs):
    c = count(xs)
    return max(c,key=lambda k : c[k])

meses_c = { mes : max_count(cs)\
           for mes,cs in meses_cs.items() }
```

CSV

- Retornar um par com o mês mais seco de 2022 e a sua previsão

```
import dateutil.parser as date

meses_date = { date.parse(mes) : c for mes,c in meses_c.items() }
meses_2022 = { mes.month : meses_date[mes] for mes in meses_date\
               if mes.year == 2022 }

def desclassifica(s):
    if   s=='chuva extrema'   : return 4
    elif s=='chuva severa'    : return 3
    elif s=='chuva moderada'  : return 2
    elif s=='chuva fraca'     : return 1
    elif s=='normal'          : return 0
    elif s=='seca fraca'      : return -1
    elif s=='seca moderada'   : return -2
    elif s=='seca severa'     : return -3
    elif s=='seca extrema'    : return -4
    else                      : return None

mais_seco_2022 = min(meses_2022.values(),key=desclassifica)
meses_mais_secos_2022 = { k for k,v in meses_2022.items()\
                          if v==mais_seco_2022 }
```

CSV

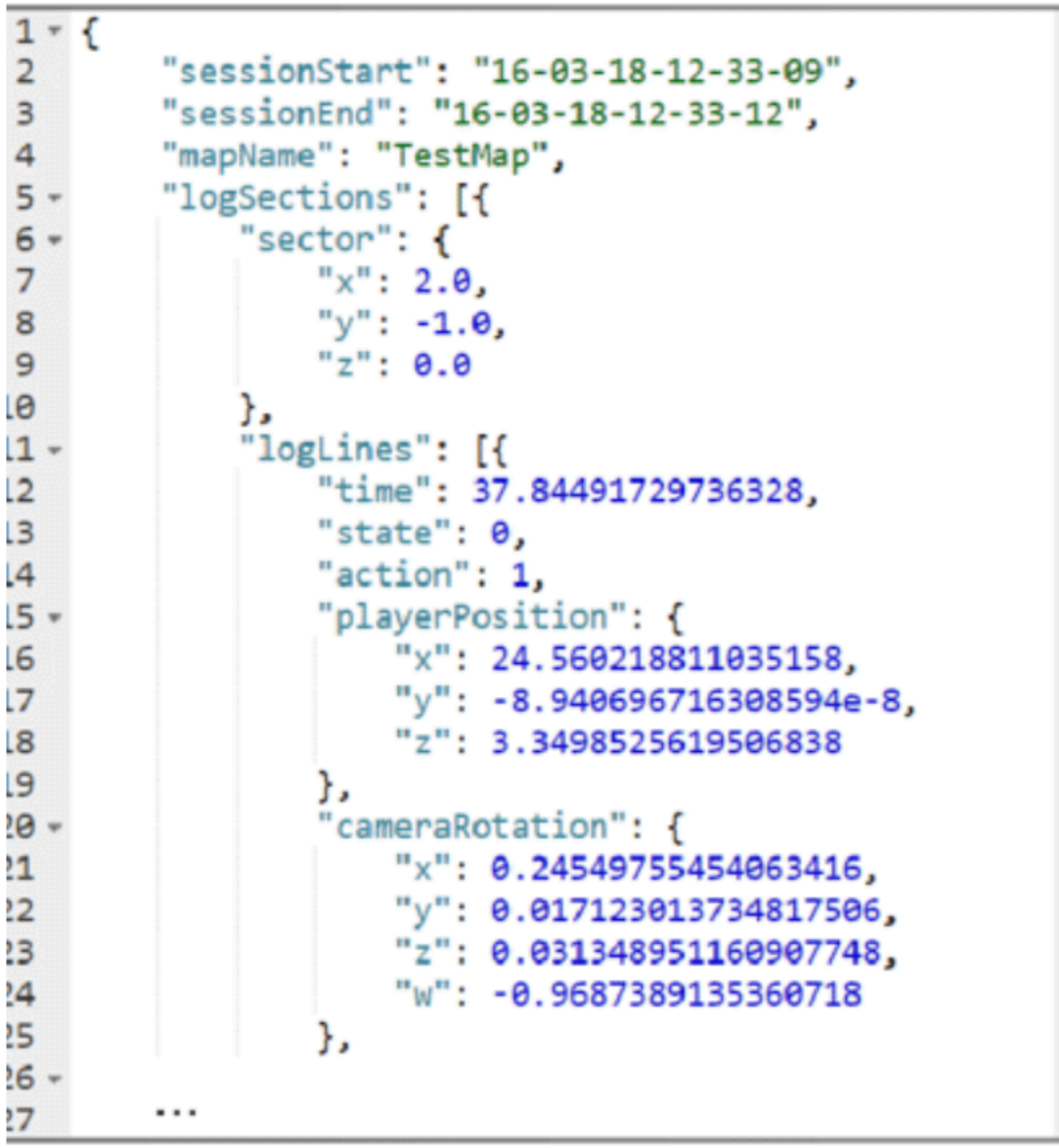
- Escrever num ficheiro CSV em Python
- E.g., guardar os índices de secura para 2022

```
meses_2022_tbl = [[‘mes’,'secura']]\
               + [ [m,s] for m,s in meses_2022.items() ]

with open('test.csv', 'w') as f:
    writer = csv.writer(f, delimiter=',')
    writer.writerows(meses_2022_tbl)
```

JSON

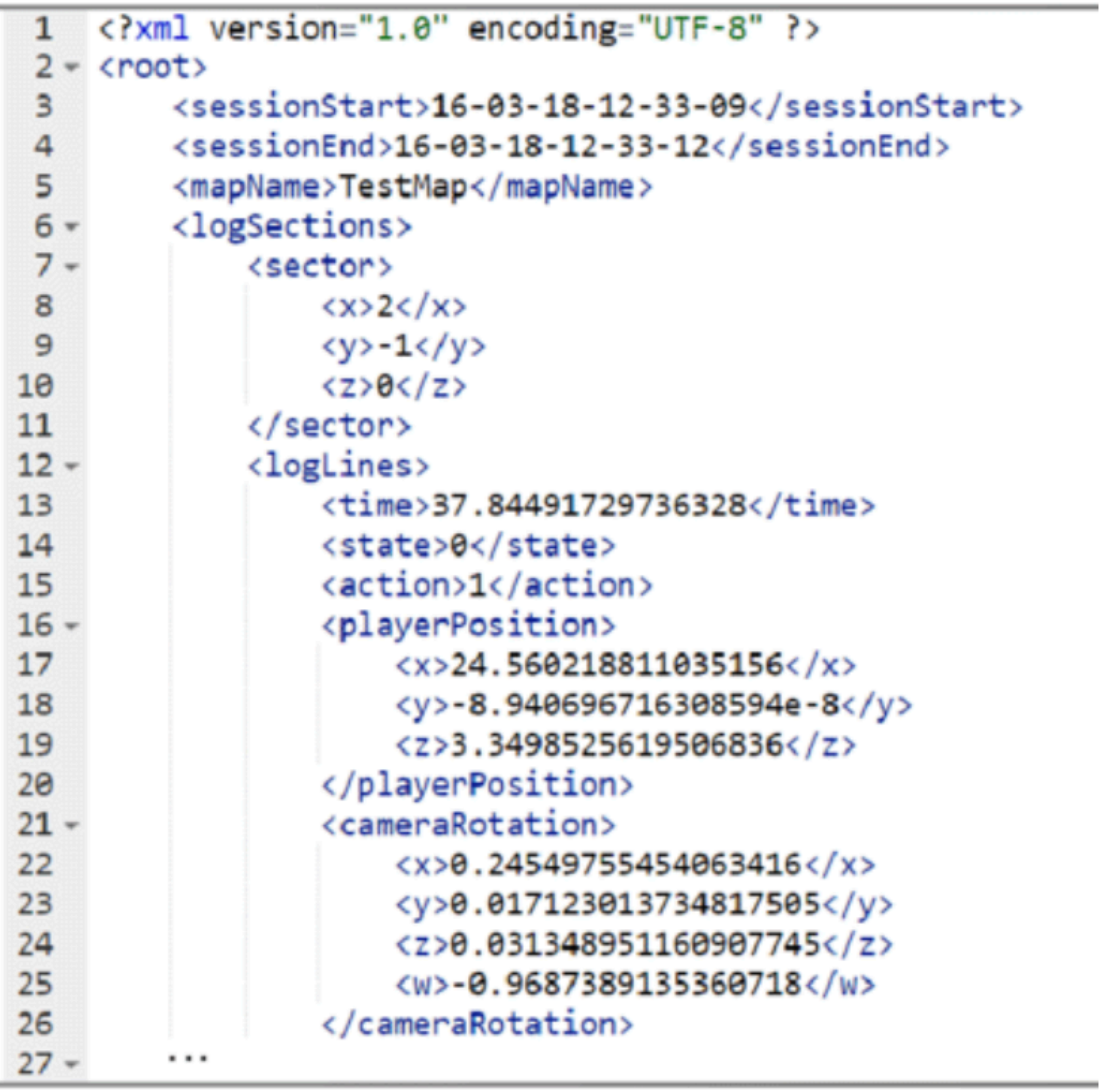
- ficheiros de texto key-value hierárquicos
- formato “standard” para troca de dados semi-estruturados entre aplicações

JSON



```json
{
    "sessionStart": "16-03-18-12-33-09",
    "sessionEnd": "16-03-18-12-33-12",
    "mapName": "TestMap",
    "logSections": [{
        "sector": {
            "x": 2.0,
            "y": -1.0,
            "z": 0.0
        },
        "logLines": [{
            "time": 37.84491729736328,
            "state": 0,
            "action": 1,
            "playerPosition": {
                "x": 24.560218811035158,
                "y": -8.940696716308594e-8,
                "z": 3.3498525619506838
            },
            "cameraRotation": {
                "x": 0.24549755454063416,
                "y": 0.017123013734817506,
                "z": 0.031348951160907748,
                "w": -0.9687389135360718
            },

    ...
```

XML



```xml
<?xml version="1.0" encoding="UTF-8" ?>
<root>
    <sessionStart>16-03-18-12-33-09</sessionStart>
    <sessionEnd>16-03-18-12-33-12</sessionEnd>
    <mapName>TestMap</mapName>
    <logSections>
        <sector>
            <x>2</x>
            <y>-1</y>
            <z>0</z>
        </sector>
        <logLines>
            <time>37.84491729736328</time>
            <state>0</state>
            <action>1</action>
            <playerPosition>
                <x>24.560218811035156</x>
                <y>-8.940696716308594e-8</y>
                <z>3.3498525619506836</z>
            </playerPosition>
            <cameraRotation>
                <x>0.24549755454063416</x>
                <y>0.017123013734817505</y>
                <z>0.031348951160907745</z>
                <w>-0.9687389135360718</w>
            </cameraRotation>
    ...
```

JSON

- Ler um ficheiro JSON em Python
- E.g., previsão metereológica de 5 dias para o Porto publicada pelo IPMA aqui
- JSON ≃ estruturas de dados Python

```python
import json
with open('1131200.json','r') as f:
    dict = json.load(f)
print(dict)
```

```json
{
   "owner": "IPMA",
   "country": "PT",
   "data": [{
      "precipitaProb": "100.0",
      "tMin": "7.5",
      "tMax": "14.5",
        ...}]
     ...
}
```

JSON

- JSON = dicionários e listas aninhados uns nos outros, cujas folhas são strings, números ou booleanos
  - <u>hierárquico</u>: estrutura aninhada, em árvore
  - <u>semi-estruturado</u>: dicionários podem ter qualquer chave/valor; strings podem representar números, datas, etc

```
json       ::= dict
dict       ::= { string : value, …}
value      ::= dict | sequence | basic
sequence   ::= [ value, … ]
basic      ::= string | number | true | false | null
```

JSON

- Obter previsão para o dia mais próximo de hoje

```
import datetime
import dateutil.parser as date

data = dict['data']
weather = { date.parse(dia[‘forecastDate’])\
          : dia['idWeatherType'] for dia in data }

hoje = datetime.datetime.today()
dia = min(weather,key=lambda d : abs(hoje-d))
previsao = weather[dia]
```

JSON

- Converter código de previsão numa descrição textual
- Descrições fornecidas pelo IPMA aqui

```
with open('weather-type-classe.json','r') as f:
    data = json.load(f)['data']

classe = { d['idWeatherType']\
         : d['descWeatherTypePT']\
          for d in data }
tempo = classe[previsao]
```

JSON

- Escrever num ficheiro JSON em Python
- Permite guardar grande parte dos objetos Python em ficheiro (serialização/deserialização)

- E.g., um dicionário *{ dia : previsão textual }*, com uma formatação especial do dia

```
import calendar
def day_month(d):
    return str(d.day)+' '+calendar.month_abbr[d.month]
weather_dif = { day_month(d) : classe[w] for d,w in
weather.items() }
print(weather_dif)

with open("test.json","w") as f:
    json.dump(weather_dif,f)
with open("test.json","r") as f:
    weather_dif2 = json.load(f)
print(weather_dif2 == weather_dif)
```