

**Relatório**

**Processamento de Linguagens**

**Aluno/os: Pedro Vieira Simões / Duarte Melo**

**Professor/es: Alberto Simões**

**Licenciatura em Engenharia de Sistemas Informáticos**

Barcelos, novembro, 2021

Resumo

Trabalho prático a pares com o objetivo de escolher um dos 3 temas indicados pelo professor, neste caso, o grupo escolheu o tema B: Comma Separated Values.

Consiste num programa Python + Ply capaz de ler um file CSV e reproduzir outputs em HTML e LaTex.

Conteúdo

[1. Introdução 5](#_Toc89039823)

[2. Expressões Regulares 6](#_Toc89039824)

[3. Outputs 8](#_Toc89039825)

# Introdução

Tal como mencionado anteriormente, este projeto visa reproduzir resultados perante o input do utilizador na consola, tendo como dados a informação registada em uma tabela CSV, tendo esta informação a cerca de vários países, indicando suas capitais, línguas e moeda.

A seleção deste tema por parte do grupo teve como fundamento a ideia de que este seria o mais fácil de ser realizado e interpretado.

O grande desafio aqui foi conseguir separar tudo por vírgulas e detetar valores dentro de aspas.

Foi criada uma classe Reader, um menu de interação e algumas funções de auxílio algorítmico.

Os dados são lidos primeiramente pelo lexer e guardados numa dictionary, onde mais tarde será usada para fazer o output de resultados.

Uma imagem com texto

Descrição gerada automaticamente

# Expressões Regulares

Decidimos usar 5 tokens diferentes, cada uma para as respetivas colunas da tabela csv e a NEWLINE para ler os “\n”



Utilizamos também 3 estados diferentes, sendo que o “country” seria o estado inicial (“INITIAL”):

Uma imagem com texto

Descrição gerada automaticamente

Utilizamos estes para perante cada leitura para cada token, avançar para o estado seguinte.

**Leitura de COUNTRY**

Para ler a coluna do país foi usada uma expressão simples r”[^,]+”, pois estando este em primeiro lugar da tabela e sendo que o mesmo é único, todo o texto até a vírgula será um país.



Após a leitura de cada palavra, é indicado o type da palavra lida e para qual estado o lexer deve saltar.

Uma imagem com texto

Descrição gerada automaticamente

**Leitura de CAPITAL E CURRENCY**

Para ler a coluna da capital e para a coluna da moeda foi usada uma expressão mais complexa, pois neste caso é possível que haja valores entre aspas, algo que temos de ter em atenção nesta entrega.



Assim é capaz de ler qualquer palavra dentro de aspas (vírgulas incluídas) ou ler qualquer palavra até à próxima vírgula (caso detete que o primeiro caracter não é ‘ ” ’).

**Leitura de LANGUAGE**

Para esta expressão não foi preciso criar uma expressão regular tão complexa, pois apenas poderia ler tudo dentro de aspas até ao fim, ou qualquer palavra sem aspas até ao ”\n”.



**Leitura de NEWLINE**

Expressão regular que lê o \n e dá pass.

Uma imagem com texto, dispositivo, contador, manómetro

Descrição gerada automaticamente

Em relação às abordagens das expressões regulares, o grupo procurou ser o mais objetivo possível de forma a arranjar a solução mais completa.

# Outputs

De forma ao programa ser o mais eficiente possível, a leitura da tabela CSV é feita uma vez apenas e guardada numa dictionary, esta foi usada para fazer os respetivos outputs

Em relação ao input do utilizador:

* Utilizador pode escolher entre 5 Opções no menu:

Uma imagem com texto

Descrição gerada automaticamente

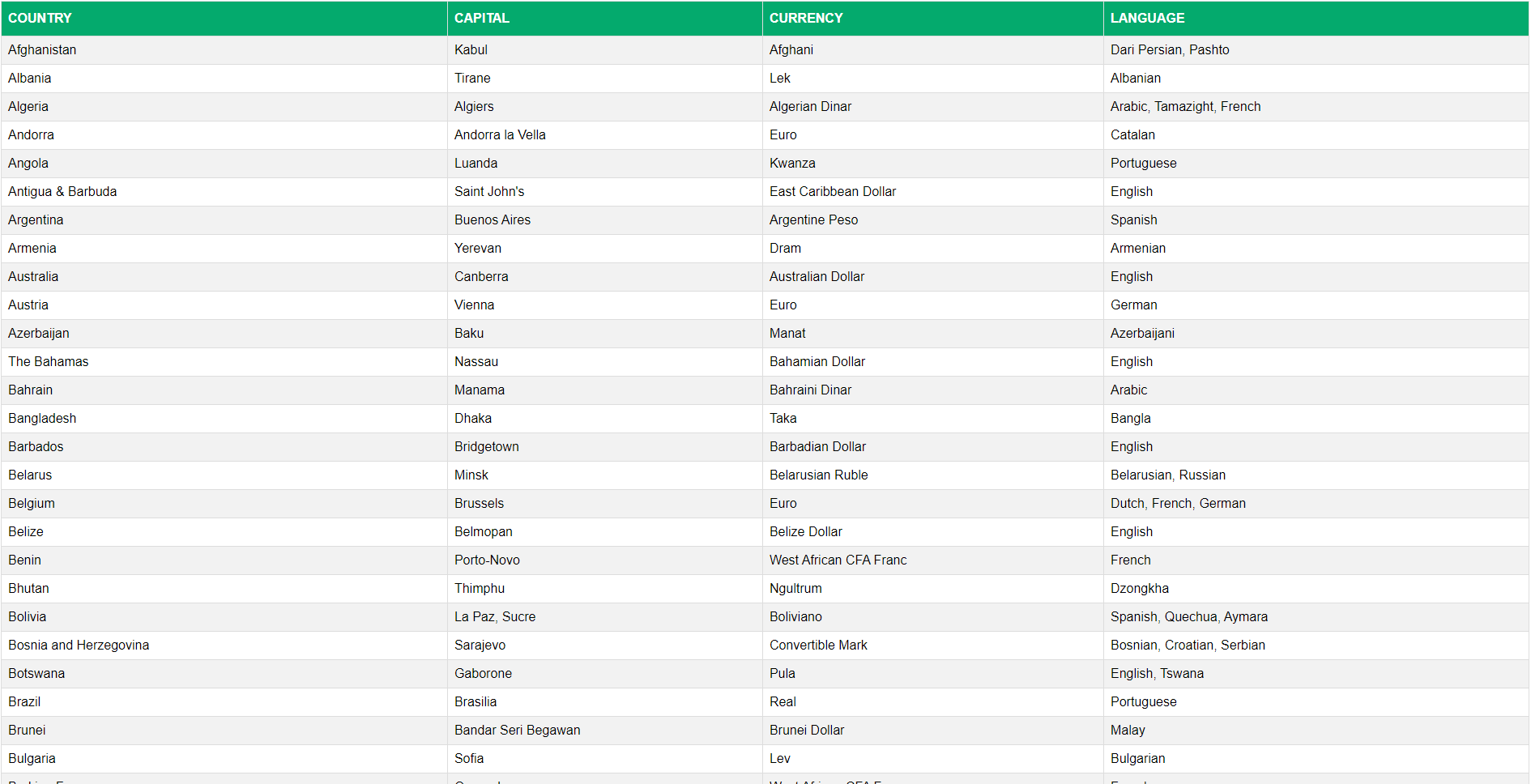
* Nas opções 1/2/3:
  + Poderá escolher de que tema pretende ver a informação.
  + Poderá também introduzir entre vírgulas 2 ou mais temas para visualizar, por exemplo: (country,capital) -> printa 2 colunas, uma dos países e outra das capitais
  + No caso de não introduzir nada, ou de um valor que não esteja incluído no cabeçalho da tabela, irá produzir o documento CSV inteiro no formato escolhido.

Uma imagem com texto

Descrição gerada automaticamente**Consola**

Inicialmente, para teste, criamos a funcionalidade de printar todos os resultados em consola. Achamos por bem, mais tarde, manter essa funcionalidade de forma que dê mais funcionalidades ao programa.

**HTML**

Reproduz uma tabela em HTML, foi acrescentado também um bocado de CSS, de forma a facilitar a leitura dos resultados. 

**LaTex**

Reproduz uma tabela em LaTex, neste caso apenas conseguimos reproduzir o texto em um ficheiro .txt mas verificamos com um compilador online que a tabela ficava bem estruturada.

