

Processamento de Linguagens e Compiladores (3º ano LCC)

Trabalho Prático 1

Relatório de Desenvolvimento

Grupo 17

Problema 3

José Pedro Gomes Ferreira
A91636

Pedro Paulo Costa Pereira
A88062

Tiago André Oliveira Leite
A91693

6 de Janeiro de 2022

Conteúdo

1	Introdução	2
2	Problema Proposto	3
2.1	Descrição	3
2.2	Requisitos	3
3	Concepção/desenho da Resolução	5
3.1	Organização e estrutura	5
3.2	GIC	5
3.3	Lexer	7
3.4	Nome dos utilizadores ordenados e respetiva entidade.	7
3.5	Entidades ordenadas e número de utilizadores registados em cada uma.	7
3.6	Distribuição dos utilizadores por nível de acesso.	8
3.7	Utilizadores agrupados por entidade, ordenados por nome e entidade.	8
3.8	Mostrar alguns indicadores.	8
3.9	Imprimir num ficheiro Json os 20 primeiros registos.	9
4	Demonstração de Funcionamento	10
4.1	Ficheiro: 'clav-users'	10
4.1.1	Conteúdo do ficheiro.	10
4.1.2	Execução e menu.	11
4.1.3	Nome dos utilizadores ordenados e respetiva entidade.	11
4.1.4	Entidades ordenadas e número de utilizadores registados em cada uma.	13
4.1.5	Distribuição dos utilizadores por nível de acesso.	13
4.1.6	Utilizadores agrupados por entidade, ordenados por nome e entidade.	15
4.1.7	Mostrar alguns indicadores.	18
4.1.8	Imprimir num ficheiro Json os 20 primeiros registos.	19
4.2	Ficheiro: 'forum-online.txt'	22
4.2.1	Conteúdo do ficheiro.	22
4.2.2	Execução e menu.	22
4.2.3	Nome dos utilizadores ordenados e respetiva entidade.	23
4.2.4	Entidades ordenadas e número de utilizadores registados em cada uma.	23
4.2.5	Distribuição dos utilizadores por nível de acesso.	24

4.2.6	Utilizadores agrupados por entidade, ordenados por nome e entidade.	25
4.2.7	Mostrar alguns indicadores.	25
4.2.8	Imprimir num ficheiro Json os 20 primeiros registos.	26
4.3	Ficheiro: 'classroom.txt'	29
4.3.1	Conteúdo do ficheiro.	29
4.3.2	Execução e menu.	31
4.3.3	Nome dos utilizadores ordenados e respetiva entidade.	31
4.3.4	Entidades ordenadas e número de utilizadores registados em cada uma.	32
4.3.5	Distribuição dos utilizadores por nível de acesso.	33
4.3.6	Utilizadores agrupados por entidade, ordenados por nome e entidade.	34
4.3.7	Mostrar alguns indicadores.	36
4.3.8	Imprimir num ficheiro Json os 20 primeiros registos.	37
5	Conclusão	41
A	Código do Programa	42

Capítulo 1

Introdução

Este documento tem como objetivo explicar a solução que implementamos para a resolução do problema proposto no âmbito da unidade curricular de Processamento de Linguagens e Compiladores.

O problema consiste em implementar uma linguagem de programação imperativa simples, com regras sintáticas definidas pelo grupo.

Para o desenvolvimento da nossa linguagem tivemos que definir uma gramática independente de contexto **GIC** e desenvolver um compilador que gera **pseudo-código**, **Assembly** para uma Máquina Virtual, VM, que nos foi fornecida.

No desenvolvimento do programa, para além do conhecimento adquirido nas aulas, utilizamos os módulos 'Yacc/ Lex' do 'PLY/Python'.

Esperamos que o resultado final cumpra todos os requisitos.

Capítulo 2

Problema Proposto

2.1 Descrição

Pretende-se que comece por definir uma linguagem de programação imperativa simples, a seu gosto. Apenas deve ter em consideração que essa linguagem terá de permitir:

- *declarar* variáveis atômicas do tipo *inteiro*, com os quais se podem realizar as habituais operações aritméticas, relacionais e lógicas;
- *efetuar* instruções algorítmicas básicas como a *atribuição do valor de expressões numéricas a variáveis*;
- *ler* do *standard input* e *escrever* no *standard output*;
- *efetuar* instruções *condicionais* para controlo do fluxo de execução;
- *efetuar* instruções *cíclicas* para controlo do fluxo de execução, permitindo o seu aninhamento.
Note que deve implementar pelo menos o ciclo **while-do**, **repeat-until** ou **for-do**.

Adicionalmente deve ainda suportar, à sua escolha, uma das duas funcionalidades seguintes:

- *declarar e manusear* variáveis estruturadas do tipo *array* (a 1 ou 2 dimensões) de inteiros, em relação aos quais é apenas permitida a operação de indexação (índice inteiro);
- *definir e invocar subprogramas* sem parâmetros mas que possam retornar um resultado do tipo inteiro.

2.2 Requisitos

- Utilização da linguagem Python.
- Resolver o problema recurso aos módulos 'Yacc/ Lex' do 'PLY/Python'.
- Gerar **pseudo-código**, **Assembly** da Máquina Virtual VM fornecida.
- Preparar um conjunto de testes de modo a ver o código Assembly gerado bem como o programa a correr na máquina virtual VM. Este conjunto terá de conter, no mínimo, os 4 primeiros exemplos abaixo e um dos 2 últimos conforme a escolha de funcionalidades da linguagem:
 - ler 4 números e dizer se podem ser os lados de um quadrado;

- ler um inteiro N , depois ler N números e escrever o menor deles;
- ler N (constante do programa) números e calcular e imprimir o seu produtório;
- contar e imprimir os números ímpares de uma sequência de números naturais;
- ler e armazenar N números num array; imprimir os valores por ordem inversa;
- invocar e usar num programa seu uma função 'potencia()', que começa por ler do input a base B e o expoente E e retorna o valor B^E .

Capítulo 3

Concepção/desenho da Resolução

3.1 Organização e estrutura

O nosso trabalho pode ser dividido em 4 partes:

- Definição da **GIC** que define a estrutura sintática da nossa linguagem.
- Construção do analisador léxico **lexer**.
- Construção do analisador sintático **parser**.
- Conversão das instruções para código **Assembly** da VM.

Todas as funções descritas neste capítulo podem ser encontradas no anexo A do documento.

No nosso trabalho optamos por implementar a funcionalidade de `textit` declarar e manusear variáveis estruturadas do tipo *array* (a 1 ou 2 dimensões) de inteiros, em relação aos quais é apenas permitida a operação de indexação (índice inteiro).

3.2 GIC

A nossa linguagem é gerada pela seguinte gramática independente de contexto:

```
program      : MAIN LCURLY body RCURLY

body         : declarations instructions

declarations :
              | declaration declarations

declaration  : type VAR SEMICOLON
              | type LBRACKET NUM RBRACKET VAR SEMICOLON
              | type LBRACKET NUM RBRACKET LBRACKET NUM RBRACKET VAR SEMICOLON

type         : INT
              | FLOAT
```

```

instructions :
| instruction instructions

instruction : atributions SEMICOLON
| WHILE LPAREN condition RPAREN LCURLY instructions RCURLY
| FOR LPAREN atributions SEMICOLON condition SEMICOLON atributions RPAREN LCURLY
  instructions RCURLY
| IF LPAREN condition RPAREN LCURLY instructions RCURLY
| IF LPAREN condition RPAREN LCURLY instructions RCURLY ELSE LCURLY
  instructions RCURLY
| SCAN LPAREN variable RPAREN SEMICOLON
| PRINT LPAREN variable RPAREN SEMICOLON
| PRINTLN LPAREN variable RPAREN SEMICOLON
| PRINT LPAREN STRING RPAREN SEMICOLON
| PRINTLN LPAREN STRING RPAREN SEMICOLON

atributions :
| atribution
| atribution COMMA atributions

atribution : variable EQUAL expression
| variable EQUAL condition

variable : VAR
| VAR LBRACKET expression RBRACKET
| VAR LBRACKET expression RBRACKET LBRACKET expression RBRACKET

expression : variable
| NUM
| REAL
| LPAREN expression RPAREN
| expression PLUS expression
| expression MINUS expression
| expression MUL expression
| expression DIV expression
| expression MOD expression

condition : expression EQEQ expression
| expression DIFF expression
| expression GREATER expression
| expression LESSER expression
| expression GREATEQ expression
| expression LESSEQ expression
| NUM
| REAL
| variable

```


3.3 Lexer

O menu do programa é implementado pela função `menu()`.

Esta função cria um ciclo `while` que mostra ao utilizador todas as opções de funcionalidades disponíveis. De seguida lê a opção escolhida pelo utilizador e caso a opção esteja entre 1 e 6 vai invocar a função que implementa a funcionalidade selecionada. Se a opção escolhida pelo utilizador for menor que 0 ou maior que 6 vai ser mostrada uma mensagem de erro sendo solicitado ao utilizador que escolha novamente. Caso a opção seja 0, o programa termina a sua execução.

Para tornar a experiencia mais agradável, entre a execução de cada funcionalidade, é feita uma limpeza do ecrã através da impressão de 50 paragrafos.

3.4 Nome dos utilizadores ordenados e respetiva entidade.

Esta funcionalidade é implementada pela função `name_entity_list()`.

Na sua implementação vamos iterar por cada linha do texto e em cada uma vamos procurar o nome de utilizador e a entidade correspondente. Para capturar o nome do utilizador vamos utilizar a função `re.match` com a expressão regular `'(\w+\.\?s*(-?\w+\.\?s*)*\b)'` e para capturar a entidade utilizamos a função `re.search` com a expressão regular `'ent_\w*'`. De seguida armazenamos o resultado num dicionario em que a chave é o nome do utilizador e o valor é uma lista com as entidades desse utilizador. Caso o utilizador ainda nao exista no dicionario é criada uma nova entrada, caso contrário a entidade capturada é acrescentada na lista do utilizador.

Por fim, transformamos o dicionario numa lista composta pelos pares (nome de utilizador, entidade(s)), ordenada por nome de utilizador, e fazemos a impressão de cada utilizador e respetiva(s) entidade(s).

3.5 Entidades ordenadas e número de utilizadores registados em cada uma.

Esta funcionalidade é implementada pela função `entity_num_elements_list()`.

Na sua implementação vamos iterar por cada linha do texto e através da função `re.search` com a expressão regular `'ent_\w*'` vamos capturar a entidade presente nessa linha. A entidade capturada é armazenada num dicionario cuja chave é o nome da entidade e o valor é o número de utilizadores nessa entidade. Caso a entidade ainda não exista no dicionario é criada uma nova entrada com o valor 1, caso contrário, o valor dessa entidade no dicionario é incrementado em 1.

Por fim, transformamos o dicionario numa lista composta pelos pares (entidade, nº de utilizadores), ordenada por entidade, e fazemos a impressão de cada entidade e respetivo nº de utilizadores.

3.6 Distribuicao dos utilizadores por nível de acesso.

Esta funcionalidade é implementada pela função `dist_users_level()`.

Na sua implementação vamos iterar por cada linha do texto e partir cada uma destas pelo separador `'[:,|]+'` com o uso da função `re.split`. De seguida vamos capturar o nome de utilizador no primeiro elemento (index 0) da linha partida com a função `re.match` e a expressão regular `'(\w+\.\?s*(-?\w+\.\?s*)*\b)'` e vamos capturar o nível de acesso no quarto elemento da linha partida (index 3) com a função `re.search` e a expressão regular `'\d+\.\?d*'`. O resultado de capturar o nível de acesso e nome de utilizador são armazenados num dicionário em que a chave é o nível de acesso e o valor é um conjunto com os utilizadores com esse nível de acesso. Optamos por um conjunto para evitar repetições, uma vez que existem utilizadores

com mais do que um nível de acesso. Caso o nível de acesso capturado não exista no dicionário, é criada uma entrada cuja chave é o nível de acesso e cujo valor é um conjunto com o utilizador capturado, caso contrário o utilizador é acrescentado ao conjunto de utilizadores já existente para aquela chave. Para ser mais fácil de contar o nº de utilizadores total utilizamos também um conjunto onde vamos colocando cada utilizador capturado.

Por fim, transformamos o dicionário numa lista composta pelos pares (nível de acesso, utilizadores) ordenada por nível de acesso e fazemos uma impressão dos utilizadores presentes em cada nível de acesso assim como a percentagem de utilizadores por nível.

3.7 Utilizadores agrupados por entidade, ordenados por nome e entidade.

Esta funcionalidade é implementada pela função `name_entity_group()`.

Na sua implementação vamos iterar por cada linha do texto e em cada uma vamos procurar o nome de utilizador e a entidade correspondente. Para capturar o nome do utilizador vamos utilizar a função `re.match` com a expressão regular `'(\w+\.\?\s*(-?\w+\.\?\s*)*\b)'` e para capturar a entidade utilizamos a função `re.search` com a expressão regular `'ent_\w*'`. De seguida armazenamos o resultado num dicionário em que a chave é a entidade e o valor é uma lista com os utilizadores dessa entidade. Caso a entidade ainda não exista no dicionário é criada uma nova entrada, caso contrário, o utilizador capturado é acrescentado na lista da entidade correspondente.

Por fim, transformamos o dicionário numa lista composta pelos pares (entidade, utilizador), ordenada por entidade, e fazemos a impressão de cada entidade e respetivos utilizadores ordenados por nome.

3.8 Mostrar alguns indicadores.

Esta funcionalidade é implementada pela função `indicators()`.

Na sua implementação vamos iterar por cada linha do texto e partir cada uma destas pelo separador `'[: ,|]+'` com o uso da função `re.split`. De seguida vamos capturar o nome de utilizador no primeiro elemento (index 0) da linha partida com a função `re.match` e a expressão regular `'(\w+\.\?\s*(-?\w+\.\?\s*)*\b)'`, vamos capturar a entidade no terceiro elemento da linha partida (index 2) com a função `re.search` e a expressão regular `'ent_\w*'`, e vamos capturar o nível de acesso no quarto elemento da linha partida (index 3) com a função `re.search` e a expressão regular `'\d+\.\?\d*'`. Os nomes de utilizadores capturados anteriormente são armazenados num conjunto, para evitar repetições, enquanto que as entidades e níveis de acesso capturados anteriormente são armazenados em dois dicionários. Num dos dicionários o par chave e valor são a entidade e respetivo número de utilizadores enquanto que no outro o par chave e valor são o nível de acesso e respetivo número de utilizadores.

Por fim, fazemos uma impressão do número total de utilizadores, o número total de entidades, o número de utilizadores por entidade e o número de utilizadores por nível de acesso.

3.9 Imprimir num ficheiro Json os 20 primeiros registos.

Esta funcionalidade é implementada pela função `json_20()`.

Na sua implementação vamos iterar por cada uma das 20 primeiras linhas do texto e partir cada uma destas pelo separador `'[: ,|]+'` com o uso da função `re.split`. De seguida vamos capturar o nome de utilizador no primeiro elemento (index 0) da linha partida com a função `re.match` e a expressão regular `'(\w+\.\?\s*(-?\w+\.\?\s*)*\b)'`, vamos capturar o email de utilizador no segundo elemento (index 1)

da linha partida com a função `re.search` e a expressão regular `'(\w+|\.|@|_|-)+'`, vamos capturar a entidade no terceiro elemento da linha partida (index 2) com a função `re.search` e a expressão regular `'ent_\w*'`, vamos capturar o nível de acesso no quarto elemento da linha partida (index 3) com a função `re.search` e a expressão regular `'\d+\.? \d*'`, e vamos capturar o número de chamadas ao backend no quinto elemento da linha partida (index 4) com a função `re.search` e a expressão regular `'\d+'`. O resultado de cada linha é armazenado numa lista de tuplos (node de utilizador, email, entidade, nível de acesso, número de chamadas ao backend).

Finalizadas as capturas pedimos ao utilizador para escolher qual o nome do ficheiro para o qual quer imprimir os dados recolhidos. Depois abrimos o ficheiro e procedemos à impressão de cada registo (tuplo armazenado na lista) num formato json.

Nota: Caso o ficheiro de input não possua 20 linhas, a função vai iterar por todas as linhas existentes no texto e apresentar os registos encontrados.

Capítulo 4

Demonstração de Funcionamento

4.1 Ficheiro: 'clav-users'

4.1.1 Conteúdo do ficheiro.

```
Élia Cristina Viegas Pedro :: epedro@ccdr-alg.pt :: ent_CCDR-Alg :: 1 :: 0
Formação DGLAB :: lurdes.almeida@dglab.gov.pt :: ent_DGLAB :: 3.5 :: 0
Nuno Filipe Casas Novas :: nuno.novas@ccdr-lvt.pt :: ent_CCDR-LVT :: 1 :: 0
Sónia Patrícia Pinheiro Reis :: sonia.reis@icnf.pt :: ent_ICNF :: 1 :: 0
Sónia Isabel Ferreira Gonçalves Negrão :: sonia.negrao@cm-albufeira.pt :: ent_CMABF :: 1 :: 0
Filipe Ferreira Cardoso Leitão :: arquivo@cm-spsul.pt :: ent_CMSPS :: 1 :: 0
Ana Lúcia Cabrita Guerreiro :: alucia@ccdr-alg.pt :: ent_CCDR-Alg :: 1 :: 0
Alda do Carmo Namora Soares de Andrade :: aandrade@letras.ulisboa.pt :: ent_FLUL :: 1 :: 0
Ricardo Almeida :: ricardo.almeida@dgeg.gov.pt :: ent_DGEG :: 1 :: 0
Sandra Cristina Patrício da Silva :: spatricio@mun-sines.pt :: ent_CMSNS :: 1 :: 0
Cátia João Matias Trindade :: catia.trindade@dglab.gov.pt :: ent_DGLAB :: 4 :: 0
Ricardo Canela :: tyty@tyty.pt :: ent_BdP :: 3 :: 0
Cátia Trindade :: matiasjcatia@gmail.com :: ent_DGLAB :: 4 :: 0
Miguel Ferreira :: mferreira@keep.pt :: ent_KEEP :: 7 :: 0
Fernando Manuel Antunes Marques da Silva :: fernando.marques.silva@marinha.pt :: ent_STI-M :: 1
Ana Maria Teixeira Gaspar :: ana.gaspar@sgmf.gov.pt :: ent_SGMF :: 1 :: 0
Maria Matos de Almeida Talhada Correia :: MariaMatos.Correia@icnf.pt :: ent_ICNF :: 1 :: 0
Cármem Isabel Amador Francisco :: carmem@mun-sines.pt :: ent_CMSNS :: 1 :: 0
Maria Leonor da Conceição Fresco Franco :: leonor.mina@ccdr-lvt.pt :: ent_CCDR-LVT :: 1 :: 0
Maria Rita Gago :: m-rita.gago@dglab.gov.pt :: ent_DGLAB :: 6 :: 3
José Carlos Leite Ramalho :: jcr@di.uminho.pt :: ent_A3ES :: 7 :: 3
Regina Neves Lopes :: Regina.Lopes@sgmf.gov.pt :: ent_SGMF :: 1 :: 0
João Paulo de Melo Esteves Pereira :: joao.pereira@apambiente.pt :: ent_APA :: 1 :: 0
António José Morim Brandão :: Antonio.brandao@metrodoporto.pt :: ent_MdP :: 1 :: 0
Alexandra Lourenço :: alexandra.lourenco@dglab.gov.pt :: ent_DGLAB :: 7 :: 3
Rui Araújo :: pg38425@alunos.uminho.pt :: ent_II :: 7 :: 9
Filipa Carvalho :: filipa.carvalho@dglab.gov.pt :: ent_DGLAB :: 4 :: 0
clara cristina rainho viegas :: clara.viegas@dglab.gov.pt :: ent_DGLAB :: 4 :: 0
Madalena Ribeiro :: madalena.ribeiro@dglab.gov.pt :: ent_DGLAB :: 6 :: 0
Zélia Gomes :: zelia.gomes@dglab.gov.pt :: ent_DGLAB :: 4 :: 0
```

```

Frederico Pinto :: frederico21pinto@hotmail.com :: ent_ACSS :: 7 :: 0
Pedro Penteado :: pedro.penteado@dglab.gov.pt :: ent_DGLAB :: 5 :: 1
Madalena Ribeiro :: madalena.ribeiro07@gmail.com :: ent_DGLAB :: 4 :: 2
Alexandre Teixeira :: a73547@alunos.uminho.pt :: ent_A3ES :: 7 :: 0
octavio :: octaviojmaia@gmail.com :: ent_A3ES :: 2 :: 0
José Carlos Martins :: a78821@alunos.uminho.pt :: ent_A3ES :: 7 :: 0
Silvestre Lacerda :: silvestre.lacerda@dglab.gov.pt :: ent_DGLAB :: 5 :: 0
Design-DGLAB :: a75536@alunos.uminho.pt :: ent_DGLAB :: 7 :: 0
João Pimentel :: A80874@alunos.uminho.pt :: ent_A3ES :: 7 :: 0
Maria Celeste Pereira :: m-celeste.pereira@dglab.gov.pt :: ent_DGLAB :: 4 :: 0
CLAV-migrator :: a74036@alunos.uminho.pt :: ent_A3ES :: 7 :: 0
Alexadre Teixeira :: alex@keep.pt :: ent_KEEP :: 7 :: 0
Carlos Barbosa :: A82324@alunos.uminho.pt :: ent_A3ES :: 7 :: 0
Rui Araújo Simples :: 0.rffa.0@gmail.com :: ent_LNEC :: 1 :: 0
Manuel Monteiro :: a74036@alunos.uminho.pt :: ent_A3ES :: 7 :: 0
Rui Araújo Entidade :: ruifilipearaujo@hotmail.com :: ent_AAN :: 1 :: 0
Duarte Freitas :: a63129@alunos.uminho.pt :: ent_A3ES :: 7 :: 0
Maria José Maciel Chaves :: m-jose.chaves@dglab.gov.pt :: ent_DGLAB :: 4 :: 0
jcm :: jcm@live.com :: ent_AAN :: 1 :: 0
José Carlos Leite Ramalho :: jcr@dglab.pt :: ent_DGLAB :: 7 :: 0
Alexandra Testes :: m-alexandra.lourenco@dglab.gov.pt :: ent_A3ES :: 1 :: 0
Paulo Lima :: plima@keep.pt :: ent_KEEP :: 7 :: 0
Alexandra Maria Alves Coutinho Rodrigues :: acoutinh@utad.pt :: ent_UTAD :: 1 :: 0
Aluno de DAW2020 :: leo.ramalho99@gmail.com :: ent_A3ES :: 2 :: 0
PRI2020-teste :: pri2020@teste.uminho.pt :: ent_A3ES :: 2 :: 0
DAW2020-teste :: daw2020@teste.uminho.pt :: ent_A3ES :: 2 :: 0
jcr-rep-entidade :: zzglider@gmail.com :: ent_A3ES :: 1 :: 0
Carlos Matoso :: cmatoso@ambisig.com :: ent_IEFP :: 2 :: 0
Joana Braga :: jbraga@ambisig.com :: ent_IEFP :: 2 :: 0
David Ferreira :: david.ferreira@ambisig.com :: ent_IEFP :: 2 :: 0

```

4.1.2 Execução e menu.

```
>> python3 main.py clav-users.txt
```

```
*** Selecione Opção ***
```

1. Listagem com o nome e a entidade do utilizador, ordenada alfabeticamente por nome.
 2. Lista ordenada alfabeticamente das entidades referenciadas, indicando, para cada uma, quantos utilizadores estão registados.
 3. Distribuição de utilizadores por níveis de acesso.
 4. Utilizadores, agrupados por entidade, ordenada primeiro pela entidade e dentro desta pelo nome.
 5. Mostrar alguns indicadores.
 6. Imprimir os 20 primeiros registos num novo ficheiro de output mas em formato json.
 0. Sair.
- ```
>>
```

#### 4.1.3 Nome dos utilizadores ordenados e respetiva entidade.

```
>> 1
```

```
*** Utilizador : Entidade(s) ***
```

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  
 Ana Maria Teixeira Gaspar : ent\_SGMF  
 António José Morim Brandão : ent\_MdP  
 Carlos Barbosa : ent\_A3ES  
 Carlos Matoso : ent\_IEFP  
 Cármen Isabel Amador Francisco : ent\_CMSNS  
 Cátia João Matias Trindade : ent\_DGLAB  
 Cátia Trindade : ent\_DGLAB  
 clara cristina rainho viegas : ent\_DGLAB  
 CLAV-migrator : ent\_A3ES  
 David Ferreira : ent\_IEFP  
 DAW2020-teste : ent\_A3ES  
 Design-DGLAB : ent\_DGLAB  
 Duarte Freitas : ent\_A3ES  
 Élia Cristina Viegas Pedro : ent\_CCDR  
 Fernando Manuel Antunes Marques da Silva : ent\_STI  
 Filipa Carvalho : ent\_DGLAB  
 Filipe Ferreira Cardoso Leitão : ent\_CMSPS  
 Formação DGLAB : ent\_DGLAB  
 Frederico Pinto : ent\_ACSS  
 jcm : ent\_AAN  
 jcr-rep-entidade : ent\_A3ES  
 Joana Braga : ent\_IEFP  
 João Paulo de Melo Esteves Pereira : ent\_APA  
 João Pimentel : ent\_A3ES  
 José Carlos Leite Ramalho : ent\_A3ES, ent\_DGLAB  
 José Carlos Martins : ent\_A3ES  
 Madalena Ribeiro : ent\_DGLAB, 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  
 Maria Matos de Almeida Talhada Correia : ent\_ICNF  
 Maria Rita Gago : ent\_DGLAB  
 Miguel Ferreira : ent\_KEEP  
 Nuno Filipe Casas Novas : ent\_CCDR  
 octavio : ent\_A3ES  
 Paulo Lima : ent\_KEEP  
 Pedro Penteado : ent\_DGLAB  
 PRI2020-teste : ent\_A3ES  
 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

Pressione Enter!

>>

#### 4.1.4 Entidades ordenadas e número de utilizadores registados em cada uma.

>> 2

\*\*\* Entidade : Nº Utilizadores \*\*\*

ent\_A3ES : 14  
ent\_AAN : 2  
ent\_ACSS : 1  
ent\_APA : 1  
ent\_BdP : 1  
ent\_CCDD : 4  
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 : 1  
ent\_UTAD : 1

Pressione Enter!

>>

#### 4.1.5 Distribuição dos utilizadores por nível de acesso.

>> 3

\*\*\* Nível de Acesso : Distribuição \*\*\*

Nível 1 : 40%

- \* Alda do Carmo Namora Soares de Andrade
- \* Alexandra Maria Alves Coutinho Rodrigues
- \* Alexandra Testes
- \* Ana Lúcia Cabrita Guerreiro
- \* Ana Maria Teixeira Gaspar
- \* António José Morim Brandão
- \* Cármen Isabel Amador Francisco
- \* Élia Cristina Viegas Pedro
- \* Fernando Manuel Antunes Marques da Silva
- \* Filipe Ferreira Cardoso Leitão
- \* jcm
- \* jcr-rep-entidade
- \* João Paulo de Melo Esteves Pereira
- \* Maria Leonor da Conceição Fresco Franco
- \* Maria Matos de Almeida Talhada Correia
- \* Nuno Filipe Casas Novas
- \* Regina Neves Lopes
- \* Ricardo Almeida
- \* Rui Araújo Entidade
- \* Rui Araújo Simples
- \* Sandra Cristina Patrício da Silva
- \* Sónia Isabel Ferreira Gonçalves Negrão
- \* Sónia Patrícia Pinheiro Reis

Nível 2 : 12%

- \* Aluno de DAW2020
- \* Carlos Matoso
- \* David Ferreira
- \* DAW2020-teste
- \* Joana Braga
- \* octavio
- \* PRI2020-teste

Nível 3 : 2%

- \* Ricardo Canela

Nível 3.5 : 2%

- \* Formação DGLAB

Nível 4 : 14%

- \* Cátia João Matias Trindade
- \* Cátia Trindade
- \* clara cristina rainho viegas
- \* Filipa Carvalho



- \* Madalena Ribeiro
- \* Maria Celeste Pereira
- \* Maria José Maciel Chaves
- \* Zélia Gomes

Nível 5 : 3%

- \* Pedro Penteado
- \* Silvestre Lacerda

Nível 6 : 3%

- \* Madalena Ribeiro
- \* Maria Rita Gago

Nível 7 : 26%

- \* Alexadre Teixeira
- \* Alexandra Lourenço
- \* Alexandre Teixeira
- \* Carlos Barbosa
- \* CLAV-migrator
- \* Design-DGLAB
- \* Duarte Freitas
- \* Frederico Pinto
- \* João Pimentel
- \* José Carlos Leite Ramalho
- \* José Carlos Martins
- \* Manuel Monteiro
- \* Miguel Ferreira
- \* Paulo Lima
- \* Rui Araújo

Pressione Enter!

>>

#### 4.1.6 Utilizadores agrupados por entidade, ordenados por nome e entidade.

>> 4

\*\*\* Utilizadores agrupados por entidade \*\*\*

ent\_A3ES:

- \* Alexandra Testes
- \* Alexandre Teixeira
- \* Aluno de DAW2020
- \* Carlos Barbosa
- \* CLAV-migrator
- \* DAW2020-teste
- \* Duarte Freitas
- \* jcr-rep-entidade

- \* João Pimentel
- \* José Carlos Leite Ramalho
- \* José Carlos Martins
- \* 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:

- \* Ana Lúcia Cabrita Guerreiro
- \* Élia Cristina Viegas Pedro
- \* 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
- \* Cátia João Matias Trindade
- \* Cátia Trindade
- \* clara cristina rainho viegas
- \* 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:

- \* Fernando Manuel Antunes Marques da Silva

ent\_UTAD:

- \* Alexandra Maria Alves Coutinho Rodrigues

Pressione Enter!

>>

#### 4.1.7 Mostrar alguns indicadores.

>> 5

\*\*\* Indicadores \*\*\*

Número de Utilizadores: 58

Número de Entidades: 21

Distribuição de utilizadores por entidade:

- \* ent\_A3ES : 14
- \* ent\_AAN : 2
- \* ent\_ACSS : 1
- \* ent\_APA : 1
- \* ent\_BdP : 1
- \* ent\_CCDD : 4
- \* 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 : 1
- \* ent\_UTAD : 1

Distribuição de utilizadores por nível:

- \* Nível 1 : 23
- \* Nível 2 : 7
- \* Nível 3 : 1
- \* Nível 3.5 : 1
- \* Nível 4 : 8
- \* Nível 5 : 2
- \* Nível 6 : 2
- \* Nível 7 : 16

Pressione Enter!

>>

#### 4.1.8 Imprimir num ficheiro Json os 20 primeiros registos.

```
>> 6
```

Digite nome do ficheiro de output!

```
>> json.txt
```

Ficheiro "json.txt" gerado com sucesso!

Pressione Enter!

```
>>
```

Conteúdo do ficheiro json.txt

```
{
 "registos":[
 {
 "utilizador":"Élia Cristina Viegas Pedro",
 "email":"epedro@ccdr-alg.pt",
 "entidade":"ent_CCDR",
 "nível de acesso":"1",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Formação DGLAB",
 "email":"lurdes.almeida@dglab.gov.pt",
 "entidade":"ent_DGLAB",
 "nível de acesso":"3.5",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Nuno Filipe Casas Novas",
 "email":"nuno.novas@ccdr-lvt.pt",
 "entidade":"ent_CCDR",
 "nível de acesso":"1",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Sónia Patrícia Pinheiro Reis",
 "email":"sonia.reis@icnf.pt",
 "entidade":"ent_ICNF",
 "nível de acesso":"1",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Sónia Isabel Ferreira Gonçalves Negrão",
 "email":"sonia.negrao@cm-albufeira.pt",
 "entidade":"ent_CMABF",
 "nível de acesso":"1",
```

```

 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Filipe Ferreira Cardoso Leitão",
 "email":"arquivo@cm-spsul.pt",
 "entidade":"ent_CMSPS",
 "nível de acesso":"1",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Ana Lúcia Cabrita Guerreiro",
 "email":"alucia@ccdr-alg.pt",
 "entidade":"ent_CCDR",
 "nível de acesso":"1",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Alda do Carmo Namora Soares de Andrade",
 "email":"aandrade@letras.ulisboa.pt",
 "entidade":"ent_FLUL",
 "nível de acesso":"1",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Ricardo Almeida",
 "email":"ricardo.almeida@dgeg.gov.pt",
 "entidade":"ent_DGEG",
 "nível de acesso":"1",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Sandra Cristina Patrício da Silva",
 "email":"spatricio@mun-sines.pt",
 "entidade":"ent_CMSNS",
 "nível de acesso":"1",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Cátia João Matias Trindade",
 "email":"catia.trindade@dglab.gov.pt",
 "entidade":"ent_DGLAB",
 "nível de acesso":"4",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Ricardo Canela",
 "email":"tyty@tyty.pt",
 "entidade":"ent_BdP",

```

```

 "nível de acesso": "3",
 "número de chamadas ao backend": "0"
 },
 {
 "utilizador": "Cátia Trindade",
 "email": "matiasjcatia@gmail.com",
 "entidade": "ent_DGLAB",
 "nível de acesso": "4",
 "número de chamadas ao backend": "0"
 },
 {
 "utilizador": "Miguel Ferreira",
 "email": "mferreira@keep.pt",
 "entidade": "ent_KEEP",
 "nível de acesso": "7",
 "número de chamadas ao backend": "0"
 },
 {
 "utilizador": "Fernando Manuel Antunes Marques da Silva",
 "email": "fernando.marques.silva@marinha.pt",
 "entidade": "ent_STI",
 "nível de acesso": "1",
 "número de chamadas ao backend": "0"
 },
 {
 "utilizador": "Ana Maria Teixeira Gaspar",
 "email": "ana.gaspar@sgmf.gov.pt",
 "entidade": "ent_SGMF",
 "nível de acesso": "1",
 "número de chamadas ao backend": "0"
 },
 {
 "utilizador": "Maria Matos de Almeida Talhada Correia",
 "email": "MariaMatos.Correia@icnf.pt",
 "entidade": "ent_ICNF",
 "nível de acesso": "1",
 "número de chamadas ao backend": "0"
 },
 {
 "utilizador": "Cármem Isabel Amador Francisco",
 "email": "carmem@mun-sines.pt",
 "entidade": "ent_CMSNS",
 "nível de acesso": "1",
 "número de chamadas ao backend": "0"
 },
 {
 "utilizador": "Maria Leonor da Conceição Fresco Franco",
 "email": "leonor.mina@ccdr-lvt.pt",

```

```

 "entidade":"ent_CCDR",
 "nível de acesso":"1",
 "número de chamadas ao backend":"0"
 },
 {
 "utilizador":"Maria Rita Gago",
 "email":"m-rita.gago@dglab.gov.pt",
 "entidade":"ent_DGLAB",
 "nível de acesso":"6",
 "número de chamadas ao backend":"3"
 }
]
}

```

## 4.2 Ficheiro: 'forum-online.txt'

### 4.2.1 Conteúdo do ficheiro.

```

registado777, registado777@gmail.com, ent_C, 2, 1
Dolores , mae_coragem@yahoo.com, ent_V, 2, 7
Prof. Marcelo, marselfie@hotmail.com, ent_P, 1, 2
CR7, siiiiiim@hotmail.com, ent_D, 7, 7
Eng. F. Santos, mete_mais_um_trinco@protonmail.com , ent_D, 3, 10
Xico Esperto, o.xico.esperto@protonmail.com, ent_C, 4 , 7
Dr. Salgado, ddt@protonmail.com, ent_D, 2, 30
Sr. Doutor, sedutor@gmail.com, ent_D, 5, 1
Passos Dias Aguiar Mota, pdamota@gmail.com, ent_C, 8, 2
JJ, peaners@hotmail.com, ent_D, 3, 88
Zé Povinho, zezinho@gmail.com, ent_C, 2, 0
Socrates , animal_feroz@yahoo.pt, ent_P, 3, 2
Inacio, naosejasinacio@hotmail.com, ent_C, 4, 2
Geremias, geremeias666@yahoo.pt, ent_C, 4, 3
ativista, a.t.i.v.i.s.t.a@hotmail.com, ent_C, 2, 3
Baltasar Sete-Sóis, b72@hotmail.com, ent_V, 6, 1
geringonça, passarola@outlook.pt, ent_P, 4, 9
abelha maia, maia_a_abelha@yahoo.com, ent_V, 3, 4
Blimunda Sete-Luas, b_7_l@gmail.com, ent_V, 8, 1
mister_vitinho, i_speak_with-the-true@gmail.com, ent_D, 4, 8
o_faneca , faneca_xpto@yahoo.com, ent_C, 2, 7
Peppa , peppa_pig@gmail.com, ent_V, 5, 1

```

### 4.2.2 Execução e menu.

```
>> python3 main.py forum-online.txt
```

```
*** Selecione Opção ***
```

```
1. Listagem com o nome e a entidade do utilizador, ordenada alfabeticamente por nome.
```



2. Lista ordenada alfabeticamente das entidades referenciadas, indicando, para cada uma, quantos utilizadores estão registados.
3. Distribuição de utilizadores por níveis de acesso.
4. Utilizadores, agrupados por entidade, ordenada primeiro pela entidade e dentro desta pelo nome.
5. Mostrar alguns indicadores.
6. Imprimir os 20 primeiros registos num novo ficheiro de output mas em formato json.
0. Sair.

>>

#### 4.2.3 Nome dos utilizadores ordenados e respetiva entidade.

>> 1

\*\*\* Utilizador : Entidade(s) \*\*\*

abelha maia : ent\_V  
 ativista : ent\_C  
 Baltasar Sete-Sóis : ent\_V  
 Blimunda Sete-Luas : ent\_V  
 CR7 : ent\_D  
 Dolores : ent\_V  
 Dr. Salgado : ent\_D  
 Eng. F. Santos : ent\_D  
 Geremias : ent\_C  
 geringonça : ent\_P  
 Inacio : ent\_C  
 JJ : ent\_D  
 mister\_vitinho : ent\_D  
 o\_faneca : ent\_C  
 Passos Dias Aguiar Mota : ent\_C  
 Peppa : ent\_V  
 Prof. Marcelo : ent\_P  
 registado777 : ent\_C  
 Socrates : ent\_P  
 Sr. Doutor : ent\_D  
 Xico Esperto : ent\_C  
 Zé Povinho : ent\_C

Pressione Enter!

>>

#### 4.2.4 Entidades ordenadas e número de utilizadores registados em cada uma.

>> 2

\*\*\* Entidade : Nº Utilizadores \*\*\*

ent\_C : 8  
 ent\_D : 6  
 ent\_P : 3  
 ent\_V : 5

Pressione Enter!

>>

#### 4.2.5 Distribuição dos utilizadores por nível de acesso.

>> 3

\*\*\* Nível de Acesso : Distribuição \*\*\*

Nível 1 : 5%

\* Prof. Marcelo

Nível 2 : 27%

\* ativista

\* Dolores

\* Dr. Salgado

\* o\_faneca

\* registado777

\* Zé Povinho

Nível 3 : 18%

\* abelha maia

\* Eng. F. Santos

\* JJ

\* Socrates

Nível 4 : 23%

\* Geremias

\* geringonça

\* Inacio

\* mister\_vitinho

\* Xico Esperto

Nível 5 : 9%

\* Peppa

\* Sr. Doutor

Nível 6 : 5%

\* Baltasar Sete-Sóis

Nível 7 : 5%

\* CR7

Nível 8 : 9%

\* Blimunda Sete-Luas

\* Passos Dias Aguiar Mota

Pressione Enter!

>>

#### 4.2.6 Utilizadores agrupados por entidade, ordenados por nome e entidade.

>> 4

\*\*\* Utilizadores agrupados por entidade \*\*\*

ent\_C:

- \* ativista
- \* Geremias
- \* Inacio
- \* o\_faneca
- \* Passos Dias Aguiar Mota
- \* registado777
- \* Xico Esperto
- \* Zé Povinho

ent\_D:

- \* CR7
- \* Dr. Salgado
- \* Eng. F. Santos
- \* JJ
- \* mister\_vitinho
- \* Sr. Doutor

ent\_P:

- \* geringonça
- \* Prof. Marcelo
- \* Socrates

ent\_V:

- \* abelha maia
- \* Baltasar Sete-Sóis
- \* Blimunda Sete-Luas
- \* Dolores
- \* Peppa

Pressione Enter!

>>

#### 4.2.7 Mostrar alguns indicadores.

>> 5

\*\*\* Indicadores \*\*\*

Número de Utilizadores: 22

Número de Entidades: 4

Distribuição de utilizadores por entidade:

- \* ent\_C : 8
- \* ent\_D : 6
- \* ent\_P : 3
- \* ent\_V : 5

Distribuição de utilizadores por nível:

- \* Nível 1 : 1
- \* Nível 2 : 6
- \* Nível 3 : 4
- \* Nível 4 : 5
- \* Nível 5 : 2
- \* Nível 6 : 1
- \* Nível 7 : 1
- \* Nível 8 : 2

Pressione Enter!

>>

#### 4.2.8 Imprimir num ficheiro Json os 20 primeiros registos.

>> 6

Digite nome do ficheiro de output!

>> json.txt

Ficheiro "json.txt" gerado com sucesso!

Pressione Enter!

>>

#### Conteúdo do ficheiro json.txt

```
{
 "registos":[
 {
 "utilizador":"registado777",
 "email":"registado777@gmail.com",
 "entidade":"ent_C",
 "nível de acesso":"2",
 "número de chamadas ao backend":"1"
 },
```

```

{
 "utilizador":"Dolores",
 "email":"mae_coragem@yahoo.com",
 "entidade":"ent_V",
 "nível de acesso":"2",
 "número de chamadas ao backend":"7"
},
{
 "utilizador":"Prof. Marcelo",
 "email":"marselfie@hotmail.com",
 "entidade":"ent_P",
 "nível de acesso":"1",
 "número de chamadas ao backend":"2"
},
{
 "utilizador":"CR7",
 "email":"siiiiiim@hotmail.com",
 "entidade":"ent_D",
 "nível de acesso":"7",
 "número de chamadas ao backend":"7"
},
{
 "utilizador":"Eng. F. Santos",
 "email":"mete_mais_um_trinco@protonmail.com",
 "entidade":"ent_D",
 "nível de acesso":"3",
 "número de chamadas ao backend":"10"
},
{
 "utilizador":"Xico Esperto",
 "email":"o.xico.esperto@protonmail.com",
 "entidade":"ent_C",
 "nível de acesso":"4",
 "número de chamadas ao backend":"7"
},
{
 "utilizador":"Dr. Salgado",
 "email":"ddt@protonmail.com",
 "entidade":"ent_D",
 "nível de acesso":"2",
 "número de chamadas ao backend":"30"
},
{
 "utilizador":"Sr. Doutor",
 "email":"sedutor@gmail.com",
 "entidade":"ent_D",
 "nível de acesso":"5",
 "número de chamadas ao backend":"1"
}

```

```

},
{
 "utilizador":"Passos Dias Aguiar Mota",
 "email":"pdamota@gmail.com",
 "entidade":"ent_C",
 "nível de acesso":"8",
 "número de chamadas ao backend":"2"
},
{
 "utilizador":"JJ",
 "email":"peaners@hotmail.com",
 "entidade":"ent_D",
 "nível de acesso":"3",
 "número de chamadas ao backend":"88"
},
{
 "utilizador":"Zé Povinho",
 "email":"zezinho@gmail.com",
 "entidade":"ent_C",
 "nível de acesso":"2",
 "número de chamadas ao backend":"0"
},
{
 "utilizador":"Socrates",
 "email":"animal_feroz@yahoo.pt",
 "entidade":"ent_P",
 "nível de acesso":"3",
 "número de chamadas ao backend":"2"
},
{
 "utilizador":"Inacio",
 "email":"naosejasinacio@hotmail.com",
 "entidade":"ent_C",
 "nível de acesso":"4",
 "número de chamadas ao backend":"2"
},
{
 "utilizador":"Geremias",
 "email":"geremeias666@yahoo.pt",
 "entidade":"ent_C",
 "nível de acesso":"4",
 "número de chamadas ao backend":"3"
},
{
 "utilizador":"ativista",
 "email":"a.t.i.v.i.s.t.a@hotmail.com",
 "entidade":"ent_C",
 "nível de acesso":"2",

```

```

 "número de chamadas ao backend":"3"
 },
 {
 "utilizador":"Baltasar Sete-Sóis",
 "email":"b72@hotmail.com",
 "entidade":"ent_V",
 "nível de acesso":"6",
 "número de chamadas ao backend":"1"
 },
 {
 "utilizador":"geringonça",
 "email":"passarola@outlook.pt",
 "entidade":"ent_P",
 "nível de acesso":"4",
 "número de chamadas ao backend":"9"
 },
 {
 "utilizador":"abelha maia",
 "email":"maia_a_abelha@yahoo.com",
 "entidade":"ent_V",
 "nível de acesso":"3",
 "número de chamadas ao backend":"4"
 },
 {
 "utilizador":"Blimunda Sete-Luas",
 "email":"b_7_l@gmail.com",
 "entidade":"ent_V",
 "nível de acesso":"8",
 "número de chamadas ao backend":"1"
 },
 {
 "utilizador":"mister_vitinho",
 "email":"i_speak_with-the-true@gmail.com",
 "entidade":"ent_D",
 "nível de acesso":"4",
 "número de chamadas ao backend":"8"
 }
]
}

```

## 4.3 Ficheiro: 'classroom.txt'

### 4.3.1 Conteúdo do ficheiro.

```

Isaac Newton | a117293940@alunos.ufic.pt | ent_G12 | 1 | 0
Confúcio | a257394827@alunos.ufic.pt | ent_G09 | 1 | 0
Cristovão Colombo | a467392412@alunos.ufic.pt | ent_G12 | 1 | 0
J. R. R. Tolkien | a635485767@alunos.ufic.pt | ent_G05 | 1 | 2

```

Albert Einstein | a946596172@alunos.ufic.pt | ent\_G14 | 1 | 1  
 Louis Pasteur | a327596378@alunos.ufic.pt | ent\_G01 | 1 | 0  
 Galileu Galilei | a706203648@alunos.ufic.pt | ent\_G18 | 1 | 0  
 Euclides de Alexandria | a954271239@alunos.ufic.pt | ent\_G07 | 1 | 2  
 Charles Darwin | a227493822@alunos.ufic.pt | ent\_G15 | 1 | 0  
 Nicolau Copérnico | a654642048@alunos.ufic.pt | ent\_G02 | 1 | 0  
 Antoine Lavoisier | a804944252@alunos.ufic.pt | ent\_G05 | 1 | 1  
 Michael Faraday | a375341811@alunos.ufic.pt | ent\_G17 | 1 | 0  
 James Clerk Maxwell | a127394553@alunos.ufic.pt | ent\_G08 | 1 | 1  
 Karl Marx | a677394914@alunos.ufic.pt | ent\_G08 | 1 | 2  
 William Shakespeare | a9155647520@alunos.ufic.pt | ent\_G02 | 1 | 0  
 Napoleão Bonaparte | a510264723@alunos.ufic.pt | ent\_G19 | 2 | 1  
 Nikola Tesla | a477497937@alunos.ufic.pt | ent\_G03 | 1 | 1  
 Platão de Atenas | a617293652@alunos.ufic.pt | ent\_G04 | 1 | 1  
 Ludwig van Beethoven | a937597685@alunos.ufic.pt | ent\_G16 | 1 | 1  
 Werner Heisenberg | a237691605@alunos.ufic.pt | ent\_G00 | 1 | 0  
 Max Planck | a657792848@alunos.ufic.pt | ent\_G04 | 1 | 0  
 Júlio César | a525495438@alunos.ufic.pt | ent\_G13 | 1 | 1  
 Sigmund Freud | a922507686@alunos.ufic.pt | ent\_G13 | 1 | 3  
 Voltaire | a114503726@alunos.ufic.pt | ent\_G19 | 2 | 1  
 Johannes Kepler | a529416472@alunos.ufic.pt | ent\_G06 | 2 | 1  
 Nicolau Maquiavel | a375513115@alunos.ufic.pt | ent\_G07 | 1 | 1  
 Vasco da Gama | a867395521@alunos.ufic.pt | ent\_G14 | 1 | 2  
 Homero | a724466884@alunos.ufic.pt | ent\_G17 | 1 | 1  
 Sidarta Gautama | a417093369@alunos.ufic.pt | ent\_G20 | 1 | 0  
 Luís Vaz de Camões | a402207424@alunos.ufic.pt | ent\_G20 | 1 | 0  
 Tony Hoare | a924364751@alunos.ufic.pt | ent\_G21 | 1 | 1  
 Edsger Dijkstra | a821467613@alunos.ufic.pt | ent\_G13 | 1 | 0  
 Marie Curie | a816385193@alunos.ufic.pt | ent\_G10 | 1 | 3  
 Madre Teresa de Calcutá | a153257485@alunos.ufic.pt | ent\_G00 | 1 | 1  
 Wolfgang Amadeus Mozart | a237292435@alunos.ufic.pt | ent\_G03 | 1 | 1  
 René Descartes | a387294424@alunos.ufic.pt | ent\_G03 | 1 | 0  
 Socrates | a705853823@alunos.ufic.pt | ent\_G19 | 2 | 1  
 Arquimedes de Siracusa | a604152532@alunos.ufic.pt | ent\_G11 | 1 | 0  
 Pierre de Fermat | a807293749@alunos.ufic.pt | ent\_G01 | 1 | 2  
 Pitágoras de Samos | a224612850@alunos.ufic.pt | ent\_G10 | 1 | 1  
 Gottfried Wilhelm Leibniz | a887193477@alunos.ufic.pt | ent\_G04 | 1 | 1  
 Carl Friedrich Gauss | a927394962@alunos.ufic.pt | ent\_G11 | 1 | 0  
 Leonhard Euler | a721472365@alunos.ufic.pt | ent\_G15 | 1 | 1  
 Joseph-Louis Lagrange | a657294866@alunos.ufic.pt | ent\_G09 | 1 | 1  
 Augustin-Louis Cauchy | a390264536@alunos.ufic.pt | ent\_G18 | 1 | 1  
 Daniel Bernoulli | a317395579@alunos.ufic.pt | ent\_G16 | 1 | 1  
 Georg Cantor | a707293654@alunos.ufic.pt | ent\_G16 | 1 | 1  
 Mahatma Gandhi | a826371829@alunos.ufic.pt | ent\_G02 | 1 | 0  
 Nelson Mandela | a735598028@alunos.ufic.pt | ent\_G21 | 1 | 3  
 Fyodor Dostoevsky | a127496527@alunos.ufic.pt | ent\_G01 | 1 | 1  
 Liev Tolstói | a582250522@alunos.ufic.pt | ent\_G06 | 2 | 3  
 Henry David Thoreau | a805315792@alunos.ufic.pt | ent\_G00 | 1 | 1



### 4.3.2 Execução e menu.

```
>> python3 main.py classroom.txt
```

```
*** Selecione Opção ***
```

1. Listagem com o nome e a entidade do utilizador, ordenada alfabeticamente por nome.
  2. Lista ordenada alfabeticamente das entidades referenciadas, indicando, para cada uma, quantos utilizadores estão registados.
  3. Distribuição de utilizadores por níveis de acesso.
  4. Utilizadores, agrupados por entidade, ordenada primeiro pela entidade e dentro desta pelo nome.
  5. Mostrar alguns indicadores.
  6. Imprimir os 20 primeiros registos num novo ficheiro de output mas em formato json.
  0. Sair.
- ```
>>
```

4.3.3 Nome dos utilizadores ordenados e respetiva entidade.

```
>> 1
```

```
*** Utilizador : Entidade(s) ***
```

```
Albert Einstein : ent_G14
Antoine Lavoisier : ent_G05
Arquimedes de Siracusa : ent_G11
Augustin-Louis Cauchy : ent_G18
Carl Friedrich Gauss : ent_G11
Charles Darwin : ent_G15
Confúcio : ent_G09
Cristovão Colombo : ent_G12
Daniel Bernoulli : ent_G16
Edsger Dijkstra : ent_G13
Euclides de Alexandria : ent_G07
Fyodor Dostoevsky : ent_G01
Galileu Galilei : ent_G18
Georg Cantor : ent_G16
Gottfried Wilhelm Leibniz : ent_G04
Henry David Thoreau : ent_G00
Homero : ent_G17
Isaac Newton : ent_G12
J. R. R. Tolkien : ent_G05
James Clerk Maxwell : ent_G08
Johannes Kepler : ent_G06
Joseph-Louis Lagrange : ent_G09
Júlio César : ent_G13
Karl Marx : ent_G08
Leonhard Euler : ent_G15
Liev Tolstói : ent_G06
Louis Pasteur : ent_G01
Ludwig van Beethoven : ent_G16
Luís Vaz de Camões : ent_G20
Madre Teresa de Calcutá : ent_G00
Mahatma Gandhi : ent_G02
```

Marie Curie : ent_G10
Max Planck : ent_G04
Michael Faraday : ent_G17
Napoleão Bonaparte : ent_G19
Nelson Mandela : ent_G21
Nicolau Copérnico : ent_G02
Nicolau Maquiavel : ent_G07
Nikola Tesla : ent_G03
Pierre de Fermat : ent_G01
Pitágoras de Samos : ent_G10
Platão de Atenas : ent_G04
René Descartes : ent_G03
Sidarta Gautama : ent_G20
Sigmund Freud : ent_G13
Socrates : ent_G19
Tony Hoare : ent_G21
Vasco da Gama : ent_G14
Voltaire : ent_G19
Werner Heisenberg : ent_G00
William Shakespeare : ent_G02
Wolfgang Amadeus Mozart : ent_G03

Pressione Enter!

>>

4.3.4 Entidades ordenadas e número de utilizadores registados em cada uma.

>> 2

*** Entidade : Nº Utilizadores ***

ent_G00 : 3
ent_G01 : 3
ent_G02 : 3
ent_G03 : 3
ent_G04 : 3
ent_G05 : 2
ent_G06 : 2
ent_G07 : 2
ent_G08 : 2
ent_G09 : 2
ent_G10 : 2
ent_G11 : 2
ent_G12 : 2
ent_G13 : 3
ent_G14 : 2
ent_G15 : 2
ent_G16 : 3

```
ent_G17 : 2
ent_G18 : 2
ent_G19 : 3
ent_G20 : 2
ent_G21 : 2
```

```
Pressione Enter!
>>
```

4.3.5 Distribuição dos utilizadores por nível de acesso.

```
>> 3
```

```
*** Nível de Acesso : Distribuição ***
```

```
Nível 1 : 90%
* Albert Einstein
* Antoine Lavoisier
* Arquimedes de Siracusa
* Augustin-Louis Cauchy
* Carl Friedrich Gauss
* Charles Darwin
* Confúcio
* Cristovão Colombo
* Daniel Bernoulli
* Edsger Dijkstra
* Euclides de Alexandria
* Fyodor Dostoevsky
* Galileu Galilei
* Georg Cantor
* Gottfried Wilhelm Leibniz
* Henry David Thoreau
* Homero
* Isaac Newton
* J. R. R. Tolkien
* James Clerk Maxwell
* Joseph-Louis Lagrange
* Júlio César
* Karl Marx
* Leonhard Euler
* Louis Pasteur
* Ludwig van Beethoven
* Luís Vaz de Camões
* Madre Teresa de Calcutá
* Mahatma Gandhi
* Marie Curie
* Max Planck
* Michael Faraday
```

- * Nelson Mandela
- * Nicolau Copérnico
- * Nicolau Maquiavel
- * Nikola Tesla
- * Pierre de Fermat
- * Pitágoras de Samos
- * Platão de Atenas
- * René Descartes
- * Sidarta Gautama
- * Sigmund Freud
- * Tony Hoare
- * Vasco da Gama
- * Werner Heisenberg
- * William Shakespeare
- * Wolfgang Amadeus Mozart

Nível 2 : 10%

- * Johannes Kepler
- * Liev Tolstói
- * Napoleão Bonaparte
- * Socrates
- * Voltaire

Pressione Enter!

>>

4.3.6 Utilizadores agrupados por entidade, ordenados por nome e entidade.

>> 4

*** Utilizadores agrupados por entidade ***

ent_G00:

- * Henry David Thoreau
- * Madre Teresa de Calcutá
- * Werner Heisenberg

ent_G01:

- * Fyodor Dostoevsky
- * Louis Pasteur
- * Pierre de Fermat

ent_G02:

- * Mahatma Gandhi
- * Nicolau Copérnico
- * William Shakespeare

ent_G03:

- * Nikola Tesla
- * René Descartes
- * Wolfgang Amadeus Mozart

ent_G04:

- * Gottfried Wilhelm Leibniz
- * Max Planck
- * Platão de Atenas

ent_G05:

- * Antoine Lavoisier
- * J. R. R. Tolkien

ent_G06:

- * Johannes Kepler
- * Liev Tolstói

ent_G07:

- * Euclides de Alexandria
- * Nicolau Maquiavel

ent_G08:

- * James Clerk Maxwell
- * Karl Marx

ent_G09:

- * Confúcio
- * Joseph-Louis Lagrange

ent_G10:

- * Marie Curie
- * Pitágoras de Samos

ent_G11:

- * Arquimedes de Siracusa
- * Carl Friedrich Gauss

ent_G12:

- * Cristovão Colombo
- * Isaac Newton

ent_G13:

- * Edsger Dijkstra
- * Júlio César
- * Sigmund Freud

ent_G14:

- * Albert Einstein

* Vasco da Gama

ent_G15:

* Charles Darwin
* Leonhard Euler

ent_G16:

* Daniel Bernoulli
* Georg Cantor
* Ludwig van Beethoven

ent_G17:

* Homero
* Michael Faraday

ent_G18:

* Augustin-Louis Cauchy
* Galileu Galilei

ent_G19:

* Napoleão Bonaparte
* Socrates
* Voltaire

ent_G20:

* Luís Vaz de Camões
* Sidarta Gautama

ent_G21:

* Nelson Mandela
* Tony Hoare

Pressione Enter!

>>

4.3.7 Mostrar alguns indicadores.

>> 5

*** Indicadores ***

Número de Utilizadores: 52

Número de Entidades: 22

Distribuição de utilizadores por entidade:

* ent_G00 : 3

```
* ent_G01 : 3
* ent_G02 : 3
* ent_G03 : 3
* ent_G04 : 3
* ent_G05 : 2
* ent_G06 : 2
* ent_G07 : 2
* ent_G08 : 2
* ent_G09 : 2
* ent_G10 : 2
* ent_G11 : 2
* ent_G12 : 2
* ent_G13 : 3
* ent_G14 : 2
* ent_G15 : 2
* ent_G16 : 3
* ent_G17 : 2
* ent_G18 : 2
* ent_G19 : 3
* ent_G20 : 2
* ent_G21 : 2
```

Distribuição de utilizadores por nível:

```
* Nível 1 : 47
* Nível 2 : 5
```

Pressione Enter!

```
>>
```

4.3.8 Imprimir num ficheiro Json os 20 primeiros registos.

```
>> 6
```

Digite nome do ficheiro de output!

```
>> json.txt
```

Ficheiro "json.txt" gerado com sucesso!

Pressione Enter!

```
>>
```

Conteúdo do ficheiro json.txt

```
{
  "registos":[
    {
      "utilizador":"Isaac Newton",
      "email":"a117293940@alunos.ufic.pt",
```

```

    "entidade":"ent_G12",
    "nível de acesso":"1",
    "número de chamadas ao backend":"0"
  },
  {
    "utilizador":"Confúcio",
    "email":"a257394827@alunos.ufic.pt",
    "entidade":"ent_G09",
    "nível de acesso":"1",
    "número de chamadas ao backend":"0"
  },
  {
    "utilizador":"Cristovão Colombo",
    "email":"a467392412@alunos.ufic.pt",
    "entidade":"ent_G12",
    "nível de acesso":"1",
    "número de chamadas ao backend":"0"
  },
  {
    "utilizador":"J. R. R. Tolkien",
    "email":"a635485767@alunos.ufic.pt",
    "entidade":"ent_G05",
    "nível de acesso":"1",
    "número de chamadas ao backend":"2"
  },
  {
    "utilizador":"Albert Einstein",
    "email":"a946596172@alunos.ufic.pt",
    "entidade":"ent_G14",
    "nível de acesso":"1",
    "número de chamadas ao backend":"1"
  },
  {
    "utilizador":"Louis Pasteur",
    "email":"a327596378@alunos.ufic.pt",
    "entidade":"ent_G01",
    "nível de acesso":"1",
    "número de chamadas ao backend":"0"
  },
  {
    "utilizador":"Galileu Galilei",
    "email":"a706203648@alunos.ufic.pt",
    "entidade":"ent_G18",
    "nível de acesso":"1",
    "número de chamadas ao backend":"0"
  },
  {
    "utilizador":"Euclides de Alexandria",

```



```

    "email": "a954271239@alunos.ufic.pt",
    "entidade": "ent_G07",
    "nível de acesso": "1",
    "número de chamadas ao backend": "2"
  },
  {
    "utilizador": "Charles Darwin",
    "email": "a227493822@alunos.ufic.pt",
    "entidade": "ent_G15",
    "nível de acesso": "1",
    "número de chamadas ao backend": "0"
  },
  {
    "utilizador": "Nicolau Copérnico",
    "email": "a654642048@alunos.ufic.pt",
    "entidade": "ent_G02",
    "nível de acesso": "1",
    "número de chamadas ao backend": "0"
  },
  {
    "utilizador": "Antoine Lavoisier",
    "email": "a804944252@alunos.ufic.pt",
    "entidade": "ent_G05",
    "nível de acesso": "1",
    "número de chamadas ao backend": "1"
  },
  {
    "utilizador": "Michael Faraday",
    "email": "a375341811@alunos.ufic.pt",
    "entidade": "ent_G17",
    "nível de acesso": "1",
    "número de chamadas ao backend": "0"
  },
  {
    "utilizador": "James Clerk Maxwell",
    "email": "a127394553@alunos.ufic.pt",
    "entidade": "ent_G08",
    "nível de acesso": "1",
    "número de chamadas ao backend": "1"
  },
  {
    "utilizador": "Karl Marx",
    "email": "a677394914@alunos.ufic.pt",
    "entidade": "ent_G08",
    "nível de acesso": "1",
    "número de chamadas ao backend": "2"
  },
  {

```

```

    "utilizador":"William Shakespeare",
    "email":"a9155647520@alunos.ufic.pt",
    "entidade":"ent_G02",
    "nível de acesso":"1",
    "número de chamadas ao backend":"0"
  },
  {
    "utilizador":"Napoleão Bonaparte",
    "email":"a510264723@alunos.ufic.pt",
    "entidade":"ent_G19",
    "nível de acesso":"2",
    "número de chamadas ao backend":"1"
  },
  {
    "utilizador":"Nikola Tesla",
    "email":"a477497937@alunos.ufic.pt",
    "entidade":"ent_G03",
    "nível de acesso":"1",
    "número de chamadas ao backend":"1"
  },
  {
    "utilizador":"Platão de Atenas",
    "email":"a617293652@alunos.ufic.pt",
    "entidade":"ent_G04",
    "nível de acesso":"1",
    "número de chamadas ao backend":"1"
  },
  {
    "utilizador":"Ludwig van Beethoven",
    "email":"a937597685@alunos.ufic.pt",
    "entidade":"ent_G16",
    "nível de acesso":"1",
    "número de chamadas ao backend":"1"
  },
  {
    "utilizador":"Werner Heisenberg",
    "email":"a237691605@alunos.ufic.pt",
    "entidade":"ent_G00",
    "nível de acesso":"1",
    "número de chamadas ao backend":"0"
  }
]
}

```

Capítulo 5

Conclusão

Com o projeto concluído esperamos ter cumprido todos os requisitos que nos foram propostos e que o nosso programa respeite os princípios ensinados na disciplina. Numa primeira abordagem tentamos fazer a captura dos nomes de utilizador com a função `findall`, no entanto não estava a ser muito fácil obter resultados na captura dos nomes de utilizador, por causa do início de linha, pelo que optamos por em todos os exercícios iterar linha a linha.

No problema que nos foi proposto, em cada uma das alíneas é nos pedido que façamos um programa, no entanto achamos que a utilização do menu, na qual podemos escolher qual a funcionalidade, torna a experiência do utilizador mais simples e agradável.

Todos concordamos que o facto de o projecto ter sido desenvolvido na linguagem 'Python' e com recurso à biblioteca 're', facilitou bastante a sua implementação.

Apêndice A

Código do Programa

Ficheiro main.py

```
import re
import unicode
import sys
import os.path

def name_entity_list():
    result = {}
    for line in text:
        user = re.match(user_er,line).group()
        entity = re.search(entity_er,line).group()
        if user not in result:
            result[user] = [entity]
        else:
            result[user].append(',')
            result[user].append(entity)
    result = list(result.items())
    result.sort(key=lambda x: unicode.unidecode(x[0].casefold()))
    print("\n*** Utilizador : Entidade(s) ***\n")
    for r in result:
        print(r[0], ":", ".join(r[1]))

def entity_num_elements_list():
    entities = {}
    for line in text:
        entity = re.search(entity_er, line).group()
        if entity not in entities:
            entities[entity] = 1
        else:
            entities[entity] +=1

    result = list(entities.items())
    result.sort()
    print("\n*** Entidade : Nº Utilizadores ***\n")
    for r in result:
        print(r[0], ":", r[1])

def dist_users_level():
    levels = {}
    users = set()
    for line in text:
        broken_line = re.split(separator_er, line)
        user = re.match(user_er, broken_line[0]).group()
        level = re.search(level_er,broken_line[3]).group()
        if level not in levels:
```

```

        levels[level] = set()
        levels[level].add(user)
    else:
        levels[level].add(user)
    users.add(user)
result = list(levels.items())
result.sort()
total_users = len(users)
print("\n*** Nível de Acesso : Distribuição ***\n")
for r in result:
    print(f"Nível {r[0]} : {round((len(r[1])/total_users)*100)}%" )
    for user in sorted(r[1], key=lambda x: unicode.unidecode(x.casefold())):
        print("*",user)
    if result.index(r) != len(result)-1:
        print("")

def name_entity_group():
    result = {}
    for line in text:
        user = re.match(user_er, line).group()
        entity = re.search(entity_er, line).group()
        if entity not in result:
            result[entity] = [user]
        else:
            result[entity].append(user)

    entities = list(result.keys())
    entities.sort()
    print("\n*** Utilizadores agrupados por entidade ***\n")
    for entity in entities:
        print(f"{entity}:")
        result[entity].sort(key=lambda x: unicode.unidecode(x.casefold()))
        for user in result[entity]:
            print("*",user)
        print("")

def indicators():
    users = set()
    entities = {}
    levels = {}
    for line in text:
        broken_line = re.split(r'[:,|]+' , line)
        users.add(re.match(user_er,broken_line[0]).group())
        entity = re.search(entity_er, broken_line[2]).group()
        level = re.search(level_er, broken_line[3]).group()

        if entity not in entities:
            entities[entity] = 1
        else:
            entities[entity] += 1
        if level not in levels:
            levels[level] = 1
        else:
            levels[level] += 1

    print("\n*** Indicadores ***\n")
    print("Número de Utilizadores:",len(users))
    print("")
    print("Número de Entidades:",len(entities.keys()))
    print("")
    print("Distribuição de utilizadores por entidade:")
    for entity in sorted(entities.keys()):
        print(f"* {entity} : {entities[entity]}")
    print("")
    print("Distribuição de utilizadores por nível:")
    for level in sorted(levels.keys()):

```

```

        print(f"* Nível {level} : {levels[level]}")

def json_20():
    list = []
    N = len(text)
    if N >= 20:
        N = 20

    for i in range(N):
        broken_line = re.split(separator_er, text[i])
        user = re.match(user_er, broken_line[0]).group()
        email = re.search(email_er, broken_line[1]).group()
        entity = re.search(entity_er, broken_line[2]).group()
        level = re.search(level_er, broken_line[3]).group()
        calls = re.search(calls_er, broken_line[4]).group()
        list.append((user,email,entity,level,calls))

    file_name = input("\nDigite nome do ficheiro de output!\n>> ")
    fp = open(file_name, 'w')

    fp.write("{\n\t\t\"registos\": [\n")
    for i in range(len(list)):
        l = list[i]
        fp.write("\t\t\t{\n")
        fp.write(f"\t\t\t\t\"utilizador\": \"{l[0]}\",\n")
        fp.write(f"\t\t\t\t\"email\": \"{l[1]}\",\n")
        fp.write(f"\t\t\t\t\"entidade\": \"{l[2]}\",\n")
        fp.write(f"\t\t\t\t\"nível de acesso\": \"{l[3]}\",\n")
        fp.write(f"\t\t\t\t\"número de chamadas ao backend\": \"{l[4]}\",\n")
        if i != 19:
            fp.write("\t\t\t\t},\n")
        else:
            fp.write("\t\t\t\t}\n")
    fp.write("\t\t]\n}\n")
    fp.close()
    print(f"\nFicheiro \"{file_name}\" gerado com sucesso!")

def menu():
    cls = lambda: print('\n' * 50)
    inputfromuser = ""
    options = []
    options.append("Listagem com o nome e a entidade do utilizador, ordenada alfabeticamente por nome.")
    options.append("Lista ordenada alfabeticamente das entidades referenciadas, indicando, para cada uma, quantos utilizadores esta")
    options.append("Distribuição de utilizadores por níveis de acesso.")
    options.append("Utilizadores, agrupados por entidade, ordenada primeiro pela entidade e dentro desta pelo nome.")
    options.append("Mostrar alguns indicadores.")
    options.append("Imprimir os 20 primeiros registos num novo ficheiro de output mas em formato json.")
    cls()
    while inputfromuser != '0':
        print("*** Selecione Opção ***\n")
        for i in range(len(options)):
            print(f"{i+1}. {options[i]}")
        print("0. Sair.")

        inputfromuser = input(">> ")
        while not inputfromuser.isdigit() or int(inputfromuser) > len(options) or int(inputfromuser) < 0:
            print("Opção Invalida!")
            inputfromuser = input(">> ")

        if inputfromuser == '0':
            continue
        elif inputfromuser == '1':
            name_entity_list()
        elif inputfromuser == '2':
            entity_num_elements_list()
        elif inputfromuser == '3':

```

```

        dist_users_level()
    elif inputfromuser == '4':
        name_entity_group()
    elif inputfromuser == '5':
        indicators()
    else:
        json_20()

    input("\nPressione Enter!\n>> ")
    cls()

if len(sys.argv)>1:
    text_source = sys.argv[1]
else:
    text_source = 'clav-users.txt'

if not os.path.exists(text_source):
    print(f"Ficheiro \"{text_source}\" não encontrado!")
    sys.exit(0)

fp = open(text_source, 'r')
text = fp.readlines()
fp.close()

calls_er = r'\d+'
entity_er = r'ent_\w*'
email_er = r'(\w+|\.|_|@|-)+',
level_er = r'\d+\.\?\d*'
separator_er = r'[:,|]+'
user_er = r'(\w+\.\?s*(-?\w+\.\?s*)*\b)',

menu()

```