

Exercício1: Alunos e Disciplinas

As inscrições em disciplinas em que alunos encontram-se inscritos estão no arquivo Inscritos.csv.

(extensão csv: arquivo texto com ';' entre cada informação na linha)

Cada linha do arquivo tem a matrícula de um aluno e os códigos das disciplinas em que o aluno está inscrito.

Exemplo: arquivo Inscritos.csv

```
1612299;ENG1000;FIS1033;MAT1161;QUI1740
1413399;ENG1000;MAT1161;MAT1260
1511188;FIS1033;MAT1161;QUI1740
1412222;MAT1161;MAT1260;QUI1740
1619999;ENG1000;MAT1260
1412222;FIS1033;MAT1161;MAT1260
1617777;MAT1260;QUI1740
1418833;ENG1000;FIS1033
1611155;MAT1260;QUI1740
1414466;FIS1033;MAT1161;MAT1260
1312211;FIS1033;QUI1740
1713300;ENG1000;FIS1033;MAT1161;MAT1260
1111188;MAT1161;MAT1260
1412200;ENG1000;MAT1260
1519911;MAT1260;QUI1740
1512211;ENG1000;MAT1161
1417700;FIS1033;QUI1740
1618811;MAT1260;QUI1740
1511100;MAT1161;MAT1260
1414400;ENG1000;MAT1161;MAT1260
```

Exercício 1.1

EX1.1:

Escreva um programa para construir um dicionário de disciplinas por aluno, onde cada item do dicionario será:

MATdoALUNO: [Lista de Disciplinas do Aluno]

Ex1.1: para a solução...

a) Você deve escrever a função *criaDicDisciplinasDoAluno* que recebe um nome de arquivo, constrói e retorna um dicionário em que cada item é:

MATdoALUNO: [Lista das disciplinas cursadas].

Nessa função:

- Criar o dicionário.
- Para cada linha do arquivo (com os dados de um aluno):
 - criar uma lista com as informações do aluno a partir da linha
 - obter a tupla (*mat_do_aluno*, *Lista_das_disciplinas*)
 - incluir um novo item no dicionário, onde a matrícula deve ser a chave:
dic[mat_do_aluno] = Lista_das_disciplinas
- Retornar o dicionário

Ex1.1: solução

```
def criaDicDisciplinasDoAluno(nomeArq):
    arq = open(nomeArq, 'r')
    dicDiscDoAluno = dict()
    for linha in arq:
        lAluno = linha.strip().split(';')
        (mAluno, lDiscAluno) = (int(lAluno[0]), lAluno[1:])
        dicDiscDoAluno[mAluno] = lDiscAluno
    arq.close()
    return dicDiscDoAluno

dDiscDoAluno = criaDicDisciplinasDoAluno('Inscritos.csv')
print('\n    ALUNO: disciplinas em que está inscrito')
for aluno in dDiscDoAluno:
    print(aluno, ":", dDiscDoAluno[aluno])
```

Ex1.1: executando...

```
ALUNO: disciplinas em que está inscrito
1612299 : ['ENG1000', 'FIS1033', 'MAT1161', 'QUI1740']
1413399 : ['ENG1000', 'MAT1161', 'MAT1260']
1511188 : ['FIS1033', 'MAT1161', 'QUI1740']
1412222 : ['FIS1033', 'MAT1161', 'MAT1260']
1619999 : ['ENG1000', 'MAT1260']
1617777 : ['MAT1260', 'QUI1740']
1418833 : ['ENG1000', 'FIS1033']
1611155 : ['MAT1260', 'QUI1740']
1414466 : ['FIS1033', 'MAT1161', 'MAT1260']
1312211 : ['FIS1033', 'QUI1740']
1713300 : ['ENG1000', 'FIS1033', 'MAT1161', 'MAT1260']
1111188 : ['MAT1161', 'MAT1260']
1412200 : ['ENG1000', 'MAT1260']
1519911 : ['MAT1260', 'QUI1740']
1512211 : ['ENG1000', 'MAT1161']
1417700 : ['FIS1033', 'QUI1740']
1618811 : ['MAT1260', 'QUI1740']
1511100 : ['MAT1161', 'MAT1260']
1414400 : ['ENG1000', 'MAT1161', 'MAT1260']
```

Exercício 1.2:

dicionário inverso de um dicionário

Exercício 1.2

Inclua no seu programa a função `criaDicAlDaDisc` para construir um dicionário inverso a partir do primeiro dicionário criado. No caso, a função deve receber um dicionário como o descrito no exercício anterior e construir um novo dicionário:

um dicionário de alunos por disciplina, onde cada item do dicionário será:

CODIGOdaDISC: [Lista de Alunos(mat) na Disciplina]

A função deve retornar o dicionário inverso.

Exercício 1.2: solução

```
def criaDicAlDaDisc(dDiscDoAl):
    dAlunosDaDisc = dict()
    for aluno in dDiscDoAl:
        for disc in dDiscDoAl[aluno]:
            lstAlunosDaDisc = dAlunosDaDisc.get(disc, [])
            lstAlunosDaDisc.append(aluno)
            dAlunosDaDisc[disc] = lstAlunosDaDisc
    return dAlunosDaDisc

dAlunosDaDisc = criaDicAlDaDisc(dDiscDoAluno)
print('\n    DISCIPLINA: alunos inscritos')
for disc in dAlunosDaDisc:
    print(disc, ":", dAlunosDaDisc[disc])
```

Exercício 1.2: executando...

DISCIPLINA: alunos inscritos

'ENG1000' : [1612299, 1413399, 1619999, 1418833,
1713300, 1412200, 1512211, 1414400]

'FIS1033' : [1612299, 1511188, 1412222, 1418833,
1414466, 1312211, 1713300, 1417700]

'MAT1161' : [1612299, 1413399, 1511188, 1412222,
1414466, 1713300, 1111188, 1512211, 1511100,
1414400]

'QUI1740' : [1612299, 1511188, 1617777, 1611155,
1312211, 1519911, 1417700, 1618811]

'MAT1260' : [1413399, 1412222, 1619999, 1617777,
1611155, 1414466, 1713300, 1111188, 1412200,
1519911, 1618811, 1511100, 1414400]

Exercício 1.3

Inclua as funções correspondentes para atualizar os dois dicionários:

- o dicionário de disciplinas por aluno
- o dicionário de alunos por disciplina

a) Dada uma lista de matriculas de estudantes que trancaram tudo.

b) Dado um dicionário de matriculas de estudantes com as disciplinas que trancaram.

Exercício 2: dicionário de dicionários

Considere o arquivo `HistoricoNotas.csv` com o histórico dos alunos. Cada linha do arquivo tem a matrícula de um aluno, os códigos das disciplinas que o aluno cursou e a nota de cada disciplina.

Exemplo: arquivo `HistoricoNotas.csv`

```
1612299;ENG1000;7,8;FIS1033;5,7;MAT1161;7,6;QUI1740;7,9
1413399;ENG1000;8,4;MAT1161;9,2;MAT1260;5,3
1511188;FIS1033;8,7;MAT1161;8,5;QUI1740;8,3
1412222;MAT1161;5,4;MAT1260;3,1;QUI1740;8,1
1619999;ENG1000;6,7;MAT1260;5,1
1412222;FIS1033;4,8;MAT1161;7,2;MAT1260;3,9
1617777;MAT1260;2,3;QUI1740;9,4
1418833;ENG1000;6,9;FIS1033;3,1
1611155;MAT1260;5,3;QUI1740;8,9
1414466;FIS1033;9,4;MAT1161;5,5;MAT1260;6,1
1312211;FIS1033;6,7;QUI1740;6,7
1713300;ENG1000;8,3;FIS1033;3,4;MAT1161;5,4;MAT1260;7,1
1111188;MAT1161;4,5;MAT1260;6,1
1412200;ENG1000;6,2;MAT1260;6,6
1519911;MAT1260;9,1;QUI1740;5,4
1512211;ENG1000;8,3;MAT1161;9,1
1417700;FIS1033;2,6;QUI1740;8,8
1618811;MAT1260;4,8;QUI1740;8,2
1511100;MAT1161;7,1;MAT1260;7,3
1414400;ENG1000;8,2;MAT1161;6,7;MAT1260;6,7
```

Nos exercícios a seguir devem ser construídos e manipulados dicionários de dicionários

Exercício 2.1

Ex2.1:Escreva um programa para construir um dicionário (de dicionários) com os históricos dos alunos.

dHistorico: dicionário em que cada elemento é:

{MatriculaAluno: dDesempenho do Aluno}, onde:
dDesempenho é um dicionário com as disciplinas cursadas e suas respectivas notas:

{ CodDisc1: NotaNaDisc1, CodDisc2: NotaNaDisc2, ...}

```
dHistorico= {MAT1:{'DISC1':NT1,'DISC2':NT2,...,'DISCn':NTn},  
             MAT2:{'DISCx':NTx,'DISCy':NTy,...,'DISCm':NTm},  
             ...  
            }
```

Ex2.1: para a solução ...

- a) Escrever a função `criaDicDisciplinasDoAluno` que recebe um nome de arquivo, constrói e retorna um dicionário em que cada item é

MATdoALUNO: {dicDesempenho(Dicionário das disciplinas cursadas com notas)}.

Nessa função:

- Criar o dicionário dos históricos `dicH`.
- Para cada linha do arquivo (com os dados de um aluno):
 - criar uma lista com as informações do aluno a partir da linha
 - obter a tupla **(mat_do_aluno, dic_do_desempenho)**, devendo para isso ser utilizada a função `criaItemAluno` descrita no item b
 - incluir um novo item no dicionário `dicHistorico`, onde a matrícula deve ser a chave:
`dicHistorico[mat_do_Aluno] = dic_do_desempenho`
- Retornar o dicionário de históricos

- b) Função `criaItemAluno(dadosUmAluno)`:

Recebe um lista com dados do aluno:

`[mat, Disc1, Nt1, Disc2, Nt2, ..., Discn, Ntn]`

Retorna uma tupla com `(mat, dicDesempenho)`

Ex2.1: solução

```
def criaItemAluno(dadosUmAluno):  
    matAluno = int(dadosUmAluno[0])  
    dDesemp = {}  
    tam = len(dadosUmAluno)  
    for i in range(1,tam,2):  
        dDesemp[dadosUmAluno[i]]=float(dadosUmAluno[i+1])  
    return (matAluno, dDesemp)  
  
def criaDicDisciplinasDoAluno(nomeArq):  
    arq = open(nomeArq,'r')  
    dHistoricos = {}  
    for linha in arq:  
        (mAluno, dDesempAluno)= criaItemAluno((linha.strip()).split(';'))  
        dHistoricos[mAluno]= dDesempAluno  
    arq.close()  
    return dHistoricos
```

Ex2.1: solução

```
dicHistorico = criaDicDisciplinasDoAluno('HistoricoNotas.csv')
print("ALUNO e seu historico")
for aluno in dicHistorico:
    print(aluno, ":", dicHistorico[aluno])
```

Ex2.1: executando...

ALUNO e seu historico

```
1612299 : {'ENG1000': 7.8, 'FIS1033': 5.7, 'MAT1161': 7.6, 'QUI1740': 7.9}
1413399 : {'ENG1000': 8.4, 'MAT1161': 9.2, 'MAT1260': 5.3}
1511188 : {'FIS1033': 8.7, 'MAT1161': 8.5, 'QUI1740': 8.3}
1412222 : {'FIS1033': 4.8, 'MAT1161': 7.2, 'MAT1260': 3.9}
1619999 : {'ENG1000': 6.7, 'MAT1260': 5.1}
1617777 : {'MAT1260': 2.3, 'QUI1740': 9.4}
1418833 : {'ENG1000': 6.9, 'FIS1033': 3.1}
1611155 : {'MAT1260': 5.3, 'QUI1740': 8.9}
1414466 : {'FIS1033': 9.4, 'MAT1161': 5.5, 'MAT1260': 6.1}
1312211 : {'FIS1033': 6.7, 'QUI1740': 6.7}
1713300 : {'ENG1000': 8.3, 'FIS1033': 3.4, 'MAT1161': 5.4, 'MAT1260': 7.1}
1111188 : {'MAT1161': 4.5, 'MAT1260': 6.1}
1412200 : {'ENG1000': 6.2, 'MAT1260': 6.6}
1519911 : {'MAT1260': 9.1, 'QUI1740': 5.4}
1512211 : {'ENG1000': 8.3, 'MAT1161': 9.1}
1417700 : {'FIS1033': 2.6, 'QUI1740': 8.8}
1618811 : {'MAT1260': 4.8, 'QUI1740': 8.2}
1511100 : {'MAT1161': 7.1, 'MAT1260': 7.3}
1414400 : {'ENG1000': 8.2, 'MAT1161': 6.7, 'MAT1260': 6.7}
```

Exercício 2.2

Ex2.2: Inclua no seu programa a função para construir um dicionário a partir do dicionário com os históricos dos alunos. Deve ser construído um dicionário de alunos com notas por disciplina, onde cada item desse dicionário será:

CODIGodaDISC: {Alu1:Nt1, Alu2:Nt2, ..., AluN:NtN}

Exemplo:

dAlunosComNotaDaDisc=

```
{ 'CODIGodaDISC1': {mAlu1: Nt1, mAlu2:Nt2, ..., mAluN: NtN}

  'CODIGodaDISC2': {mAlux: Ntx, ..., mAluy: Nty} ,

                                ....

  'CODIGodaDISCw': {mAlu1: Nt1, mAlux:Ntx, ..., mAluN: NtN}

}
```


Exercício 2.2 : solução

```
def criaDicAlNtDaDisc(dHistorico):
    dDisc = {}
    for al in dHistorico:
        for disc in dHistorico[al]:
            if disc not in dDisc:
                dDisc[disc]={al:dHistorico[al][disc]}
            else:
                dDisc[disc].update({al:dHistorico[al][disc]})
    return dDisc

dAlunoseNotasDaDisc = criaDicAlNtDaDisc(dicHistorico)
print('\n DISCIPLINA e seus alunos')
for disc in dAlunoseNotasDaDisc:
    print(disc, ":", dAlunoseNotasDaDisc[disc])
```

Exercício 2.2 : executando...



DISCIPLINA e seus alunos

```
ENG1000 : {1612299: 7.8, 1413399: 8.4, 1619999: 6.7, 1418833: 6.9,  
          1713300: 8.3, 1412200: 6.2, 1512211: 8.3, 1414400: 8.2}  
FIS1033 : {1612299: 5.7, 1511188: 8.7, 1412222: 4.8, 1418833: 3.1,  
          1414466: 9.4, 1312211: 6.7, 1713300: 3.4, 1417700: 2.6}  
MAT1161 : {1612299: 7.6, 1413399: 9.2, 1511188: 8.5, 1412222: 7.2,  
          1414466: 5.5, 1713300: 5.4, 1111188: 4.5, 1512211: 9.1, 1511100:  
          7.1, 1414400: 6.7}  
QUI1740 : {1612299: 7.9, 1511188: 8.3, 1617777: 9.4, 1611155: 8.9,  
          1312211: 6.7, 1519911: 5.4, 1417700: 8.8, 1618811: 8.2}  
MAT1260 : {1413399: 5.3, 1412222: 3.9, 1619999: 5.1, 1617777: 2.3,  
          1611155: 5.3, 1414466: 6.1, 1713300: 7.1, 1111188: 6.1, 1412200: 6.6,  
          1519911: 9.1, 1618811: 4.8, 1511100: 7.3, 1414400: 6.7}
```