



Instituto Politécnico da Guarda
Escola Superior de Tecnologia e Gestão

Orientação Tutorial
Algoritmos e Programação em Python

22 novembro 2017

Trabalho de grupo | Group work
Rascunho / Draft

Curso: Engenharia Informática
Unidade Curricular:
Algoritmos e Estruturas de Dados
Algorithms and Data Structures
Ano Letivo: 2017/2018
Docente: Paulo Jorge Costa Nunes
Coordenador da área disciplinar: Noel Lopes

Conteúdo

1	Trabalho de Grupo Group Work	2
1.1	O grupo The group	2
1.2	Proposta de problema Problem statement	2
1.2.1	Disciplinas alunos	2
1.3	Objectivos	4
1.3.1	Algoritmo para calcular a qualidade de um horário	4
1.3.2	Objetivos secundários (três da lista)	5
1.4	Estruturas de dados a utilizar no trabalho	6
1.5	Acompanhamento do desenvolvimento	6
1.5.1	Métricas do código	6
1.6	Entrega, apresentação e avaliação	6
1.7	Dados Dataset	6
1.7.1	XML to Text	8
1.7.2	XML to SQL	10
1.7.3	Alunos	11
1.7.4	Disciplinas	12
1.7.5	Disciplinas alunos	12

Capítulo 1

Trabalho de Grupo | Group Work

1.1 O grupo | The group

Regras | Rules:

1. Cada grupo pode ser formado por até três alunos.
2. Cada grupo deve ter um nome. Após a formação do grupo os seus elementos devem definir um nome para o grupo. O processo para a definição do nome pode ser baseado no mesmo processo efetuado para criar uma nova empresa. O nome da empresa tem de ser deferente dos nomes das empresas já existentes.
3. Devem registar o grupo e os seus elementos no website URL antes de 1 de dezembro de 2017.

1.2 Proposta de problema | Problem statement

Com base num conjunto de dados sobre cursos, disciplinas, horário de disciplinas, salas entre outros elaborar um programa em Python que permita gerir dados de alunos (isto é: inserir, alterar, eliminar, listar, pesquisar, ordenar, etc.) e as disciplinas às quais está inscrito.

Com base nos dados pretende-se obter o horário de cada aluno num formato A4 e num formato de bolso (tamanho cartão ao de crédito).

1.2.1 Disciplinas alunos

Tabela 1.1: Horário tipo cartão de crédito

Horas	Seg	Ter	Qua	Qui	Sex
09:00-09:30	AED-T2		BD		QTS
09:30-10:00	AED-T2	PI	BD		QTS
10:00-10:30	AED-T2	PI	BD		QTS
10:30-11:00	AED-T2	PI	BD		QTS
11:00-11:30	AED-T2	PI			QTS
11:30-12:00	AED-T1	PI			QTS
12:00-12:30	AED-T1	PI			QTS
12:30-13:00	AED-T1		MG		QTS
13:00-13:30	AED-T1		MG		
Almoço					
14:00-14:30	BD			MG	
14:30-15:00	BD			MG	
15:00-15:30	BD				
15:00-16:00	ATM				
16:00-16:30	ATM				

Horário de: Paulo Nunes

Ano letivo 2017/2018
Semestre 1

Horas	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira	Sábado	Domingo
08:00 - 08:30							
08:30 - 09:00	AED-EI OT-T2 Sala: 49		BD-CM OT-1 Sala: 49		PA PA PA PA PA PA PA		
09:00 - 09:30		PI-DAI TP-1 Sala: 47					
09:30 - 10:00							
10:00 - 10:30							
10:30 - 11:00							
11:00 - 11:30							
11:30 - 12:00				AED-EI OT-T1 Sala: 49			
12:00 - 12:30							
12:30 - 13:00							
13:00 - 13:30							
13:30 - 14:00							
14:00 - 14:30	BD-CM TP-1 Sala: 49		DP DP DP DP DP DP DP DP DP DP	ATM ATM ATM			
14:30 - 15:00							
15:00 - 15:30							
15:30 - 16:00							
16:00 - 16:30							
16:30 - 17:00							
17:00 - 17:30							
17:30 - 18:00							
18:00 - 18:30							
18:30 - 19:00							
19:00 - 19:30							
19:30 - 20:00							
20:00 - 20:30							

Disciplinas

Código	Designação
AED	Algoritmos e Estruturas de Dados
BD	Bases de Dados
PI	Programação para a Internet
ATM	Atendimento
PA	Preparação de Aulas
DP	Desenvolvimento de Programas

Cursos

Sigla	Nome
EI	Engenharia Informática
CM	Comunicação Multimédia
DAI	Desenvolvimento de Aplicações Informáticas

Figura 1.1: Horário A4

O **grande desafio neste trabalho** será inventar um algoritmo que permita medir a qualidade (ex: número 1 a 10, ou texto: Muito bom, Bom, Mau, Péssimo, etc) dos horários dos alunos.

1.3 Objectivos

Desenvolver um programa em Python para gerir os horários dos alunos do IPG.

1. Definir as funcionalidades do programa (Menus / Sub-menus). As mínimas são:
 - (a) Pesquisar (nome) e listar docente
 - (b) Listar horário de docente
 - (c) Pesquisar (nome, curso) e listar disciplina
 - (d) Listar horário disciplina
 - (e) Gerir Alunos (inserir, alterar, eliminar, pesquisar) (Devem ser definidos os dados necessários)
 - (f) Gerir Disciplinas-Alunos (inserir, alterar, eliminar, pesquisar) (Devem ser definidos os dados necessários)
 - (g) Gerir Disciplinas-Horários (inserir, alterar, eliminar, pesquisar) (Os dados já existem)
 - (h) Listar horário aluno (Texto, HTML Markup Language (table, tr, th, td) ¹)
 - i. Formato Cartão de crédito
 - ii. Formato A4
 - (i) Algoritmo para calcular a qualidade de um horário (ver 1.3.1)
2. Converter os dados (1.7) XML ² para ficheiros de texto CSV ³
3. Converter os ficheiros de texto CSV para ficheiros binários
4. Desenvolver um ou mais algoritmos para resolver cada uma das funcionalidades
5. Implementar os algoritmos do item anterior em Python.

1.3.1 Algoritmo para calcular a qualidade de um horário

Para o calculo da qualidade do horário ter em conta:

Número de deslocações à Escola

Tempo para almoço

Horas de entradas e saída

Tardes livres

Manhãs livres

Sexta-feira à tarde livre

Segunda feira-manhã livre

"Furos" Número de disciplinas por dia/manhã/tarde

Número horas de aula por dia/manhã/tarde

Hora de entrada: manhã, tarde

Hora de saída: manhã, tarde

Dica: Atribuir valor de importância para cada item.

¹<https://pt.wikipedia.org/wiki/HTML>

²<https://pt.wikipedia.org/wiki/XML>

³https://pt.wikipedia.org/wiki/Comma-separated_values

1.3.2 Objetivos secundários (três da lista)

- Quantas horas de aulas um aluno tem num dia.
- Quantas horas de aulas tem um dado aluno em cada dia da semana.
- Quantas horas de aulas tem um aluno por semana.
- Quantas disciplinas tem um aluno.
- Quantos professores tem um determinado aluno.
- Quantos alunos tem um determinado professor.
- Quantos alunos têm aulas num dado dia.
- Que professores lecionam um dado número de disciplinas ao mesmo aluno.
- Que alunos estão a ter aulas num dado dia e intervalo de horas.
- Qual o aluno que tem mais horas de aulas.
- Qual o aluno que tem menos horas de aulas.
- Quais os alunos têm mais de um determinado número de disciplinas.
- Outras.
- How many hours of classes a student has in a day.
- How many hours of classes a given student has each day of the week.
- How many hours of classes a student has a week.
- How many courses does a student.
- How many teachers have a particular student.
- How many students have a particular teacher.
- How many students have classes on a given day.
- What teachers teach a given number of disciplines to the same student.
- What students are taking classes in a given day and time range.
- What is the student who has more hours of classes .
- Which the student has fewer teaching hours .
- Which students have more than a certain number of disciplines .
- Other.

1.4 Estruturas de dados a utilizar no trabalho

Elementos a utilizar no trabalho:

- Vetores, matrizes, listas, etc.
- Ficheiros: Binários para armazenar os dados da aplicação. Texto para apresentar resultados.
- Algoritmos de ordenação.
- Algoritmos de pesquisa.

1.5 Acompanhamento do desenvolvimento

Criar conta no Github: <https://github.com/>

1.5.1 Métricas do código

1.6 Entrega, apresentação e avaliação

A entrega do trabalho será efetuado através da plataforma *Moodle do IPG* ⁴, **Entrega de trabalho de grupo.**

A data de entrega é até ao **dia 8 de janeiro de 2018 até às 8H00.**

Itens a incluir na apresentação:

1. Apresentação do problema
2. Algoritmos
3. Programas em Python
 - (a) Estrutura: programa e menus
 - (b) Funcionalidades
 - (c) Demonstração
 - Capturas de ecrã (Screenshots) para cada uma das funcionalidades
 - OU**
 - Vídeo para cada uma das funcionalidades.

Os alunos para obterem nota no trabalho têm obrigatoriamente de apresentar o seu trabalho na aula do dia 20 de novembro de 2017. A ordem de apresentação dos trabalhos é de acordo com a data/hora de entrega.

A tabela 1.2 lista os item a avaliação e a sua respectiva pontuação.

1.7 Dados | Dataset

Listing 1.1: XML

```

1 <?xml version="1.0" encoding="windows-1252"?>
2 <timetable ascttversion="2018.7.3" importtype="database" options="idprefix:MyApp"
   displayname="MyApp sample XML configuration" displaycountries="us">
3   <teachers options="import:disable,canadd,canremove,canupdate,primarytt,silent" columns=
     "id,name,short">
4     <teacher id="317" name="Paulo Jorge Costa Nunes" short="PJCN"/>

```

⁴<http://moodle.ipg.pt/>

Tabela 1.2: Grelha de avaliação

Item	Classificação	Classificação aluno 1	Classificação aluno 2
Registrar do grupo e os seus elementos	1 - até à data prevista, 0 - depois		
Pergunta nº1	1		
Pergunta nº2	1		
Apresentação - dos alunos	5		
Apresentação - conteúdo	-		
1. Apresentação do problema	1		
2. Algoritmos (1/cada)	10		
3. <i>Programas em Python</i>	-		
3.1. Funcionalidades: menus	1		
3.2. Funções/Procedimentos (1/cada)	10		
4. <i>Demonstração</i>	-		
4.1 Capturas de ecrã / Vídeo	5		
4.2 Tempo real	5		
5. Qualidade de um horário	10		

```

5      <teacher id="389" name="José Alberto Quitério Figueiredo" short="JAQF"/>
6    </teachers>
7    <classes options="import:disable,canadd,canremove,canupdate,primarytt,silent" columns="
8      id,name">
9      <class id="294" name="LC1"/>
10     <class id="295" name="LC2"/>
11     <class id="296" name="LC3"/>
12     <class id="300" name="LDE1"/>
13     <class id="301" name="LDE2"/>
14   </classes>
15   <subjects options="import:disable,canadd,canremove,canupdate,primarytt,silent" columns="
16     id,name,short">
17     <subject id="INF1052_TC" name="INF1052(TC) - Algoritmos e Estruturas de Dados" short
18       ="A(TC)"/>
19     <subject id="INF1052_OT" name="INF1052(OT) - Algoritmos e Estruturas de Dados" short
20       ="A(OT)"/>
21     <subject id="INF1052_TP" name="INF1052(TP) - Algoritmos e Estruturas de Dados" short
22       ="A(TP)"/>
23     <subject id="INF1056_OT" name="INF1056(OT) - Bases de Dados I" short="BI(OT)"/>
24     <subject id="INF1056_TP" name="INF1056(TP) - Bases de Dados I" short="BI(TP)"/>
25     <subject id="INF1063_OT" name="INF1063(OT) - Programação" short="P(OT)"/>
26     <subject id="INF1063_L" name="INF1063(L) - Programação" short="P(L)"/>
27     <subject id="INF1063_TP" name="INF1063(TP) - Programação" short="P(TP)"/>
28   </subjects>
29   <classrooms options="import:disable,canadd,canremove,canupdate,primarytt,silent"
30     columns="id,name,short">
31     <classroom id="67" name="ESTG_44" short="44"/>
32     <classroom id="68" name="ESTG_45" short="45"/>
33     <classroom id="69" name="ESTG_46" short="46"/>
34     <classroom id="70" name="ESTG_47" short="47"/>
35     <classroom id="71" name="ESTG_48" short="48"/>
36     <classroom id="72" name="ESTG_49" short="49"/>
37   </classrooms>
38   <lessons options="import:disable,canadd,canremove,canupdate,primarytt,silent" columns="
39     classids,subjectid,periodsperweek,teacherids">
40     <lesson classids="268" subjectid="INF1052_OT" periodsperweek="4.0" teacherids="317"/>
41     <lesson classids="267" subjectid="INF1052_OT" periodsperweek="4.0" teacherids="317"/>
42     <lesson classids="270" subjectid="INF1007_TP" periodsperweek="4.0" teacherids="256"/>
43     <lesson classids="270" subjectid="INF1007_TP" periodsperweek="4.0" teacherids="256"/>
44     <lesson classids="270" subjectid="INF1007_TP" periodsperweek="4.0" teacherids="256"/>
45   </lessons>

```



```

39 <cards options="import:disable,canadd,canremove,canupdate,primarytt,silent" columns="
    day,period,classids,subjectid,teacherids,classroomids">
40 <card classids="268" subjectid="INF1052_OT" teacherids="317" classroomids="73" day="
    0" period="2"/>
41 <card classids="268" subjectid="INF1052_OT" teacherids="317" classroomids="73" day="
    0" period="3"/>
42 <card classids="268" subjectid="INF1052_OT" teacherids="317" classroomids="73" day="
    0" period="4"/>
43 <card classids="268" subjectid="INF1052_OT" teacherids="317" classroomids="73" day="
    0" period="5"/>
44 <card classids="267" subjectid="INF1052_OT" teacherids="317" classroomids="73" day="
    0" period="6"/>
45 <card classids="267" subjectid="INF1052_OT" teacherids="317" classroomids="73" day="
    0" period="7"/>
46 <card classids="267" subjectid="INF1052_OT" teacherids="317" classroomids="73" day="
    0" period="8"/>
47 <card classids="267" subjectid="INF1052_OT" teacherids="317" classroomids="73" day="
    0" period="9"/>
48 <card classids="268" subjectid="UINF1066_OT" teacherids="389" classroomids="51" day="
    0" period="6"/>
49 <card classids="268" subjectid="UINF1066_OT" teacherids="389" classroomids="51" day="
    0" period="7"/>
50 <card classids="268" subjectid="UINF1066_OT" teacherids="389" classroomids="51" day="
    0" period="8"/>
51 <card classids="268" subjectid="UINF1066_OT" teacherids="389" classroomids="51" day="
    0" period="9"/>
52 <card classids="267" subjectid="UINF1066_OT" teacherids="389" classroomids="51" day="
    0" period="12"/>
53 <card classids="267" subjectid="UINF1066_OT" teacherids="389" classroomids="51" day="
    0" period="13"/>
54 <card classids="267" subjectid="UINF1066_OT" teacherids="389" classroomids="51" day="
    0" period="14"/>
55 <card classids="267" subjectid="UINF1066_OT" teacherids="389" classroomids="51" day="
    0" period="15"/>
56 <card classids="268" subjectid="UINF1066_TP" teacherids="389" classroomids="51" day="
    1" period="1"/>
57 <card classids="268" subjectid="UINF1066_TP" teacherids="389" classroomids="51" day="
    1" period="2"/>
58 <card classids="268" subjectid="UINF1066_TP" teacherids="389" classroomids="51" day="
    1" period="3"/>
59 <card classids="268" subjectid="UINF1066_TP" teacherids="389" classroomids="51" day="
    1" period="4"/>
60 <card classids="267" subjectid="UINF1066_TP" teacherids="389" classroomids="51" day="
    1" period="5"/>
61 <card classids="267" subjectid="UINF1066_TP" teacherids="389" classroomids="51" day="
    1" period="6"/>
62 <card classids="267" subjectid="UINF1066_TP" teacherids="389" classroomids="51" day="
    1" period="7"/>
63 <card classids="267" subjectid="UINF1066_TP" teacherids="389" classroomids="51" day="
    1" period="8"/>
64 <card classids="268" subjectid="UINF1066_P" teacherids="389" classroomids="51" day="
    3" period="2"/>
65 <card classids="268" subjectid="UINF1066_P" teacherids="389" classroomids="51" day="
    3" period="3"/>
66 <card classids="268" subjectid="UINF1066_P" teacherids="389" classroomids="51" day="
    3" period="4"/>
67 <card classids="268" subjectid="UINF1066_P" teacherids="389" classroomids="51" day="
    3" period="5"/>
68 <card classids="267" subjectid="UINF1066_P" teacherids="389" classroomids="51" day="
    3" period="6"/>
69 <card classids="267" subjectid="UINF1066_P" teacherids="389" classroomids="51" day="
    3" period="7"/>
70 <card classids="267" subjectid="UINF1066_P" teacherids="389" classroomids="51" day="
    3" period="8"/>
71 <card classids="267" subjectid="UINF1066_P" teacherids="389" classroomids="51" day="
    3" period="9"/>
72 </cards>
73 </timetable>

```

1.7.1 XML to Text

```

1
2 import xml.etree.ElementTree as ET

```

```

3 import re
4 import os
5 file = "horario_paulo Nunes_ESTG_2017-2018.xml"
6 file = 'horario_paulo Nunes_ESTG_2017-2018_some_data.xml'
7
8 #tree = ET.parse(file)
9 #root = tree.getroot()
10 #for node in root:
11 #    print(node)
12 # <Element 'teachers' at 0x026DF8D0>
13 # <Element 'classes' at 0x026F96F0>
14 # <Element 'subjects' at 0x02700360>
15 # <Element 'classrooms' at 0x02809810>
16 # <Element 'lessons' at 0x02815C30>
17 # <Element 'cards' at 0x0284A270>
18 #exit()
19 def ConvertXMLCardsToCSV(entity):
20     print("# Entidade | Entity: %s" % entity)
21     tree = ET.parse(file)
22     root = tree.getroot()
23     es = root.find(entity)
24     print(len(es))
25     csv_all = ""
26     heads = ""
27     csv = ""
28     for e in es:
29         for k in e.keys():
30             heads = heads + "%s;" % k
31         break
32     heads = heads.rstrip(';')
33     print(heads)
34     for e in es:
35         csv = ''
36         for k in e.keys():
37             csv = csv + "%s;" % (e.get(k))
38         csv = csv.rstrip(';')
39         print (csv)
40
41 tree = ET.parse(file)
42 root = tree.getroot()
43 for node in root:
44     ConvertXMLCardsToCSV(node.tag)
45 >>>
46 # Entidade | Entity: teachers
47 2
48 short;name;id
49 PJC�;Paulo Jorge Costa Nunes;317
50 JAQF;José Alberto Quitério Figueiredo;389
51 # Entidade | Entity: classes
52 5
53 name;id
54 LC1;294
55 LC2;295
56 LC3;296
57 LDE1;300
58 LDE2;301
59 # Entidade | Entity: subjects
60 8
61 short;name;id
62 A(TC);INF1052(TC) - Algoritmos e Estruturas de Dados;INF1052_TC
63 A(OT);INF1052(OT) - Algoritmos e Estruturas de Dados;INF1052_OT
64 A(TP);INF1052(TP) - Algoritmos e Estruturas de Dados;INF1052_TP
65 BI(OT);INF1056(OT) - Bases de Dados I;INF1056_OT
66 BI(TP);INF1056(TP) - Bases de Dados I;INF1056_TP
67 P(OT);INF1063(OT) - Programação;INF1063_OT
68 P(L);INF1063(L) - Programação;INF1063_L
69 P(TP);INF1063(TP) - Programação;INF1063_TP
70 # Entidade | Entity: classrooms
71 6
72 short;name;id
73 44;ESTG_44;67
74 45;ESTG_45;68
75 46;ESTG_46;69
76 47;ESTG_47;70
77 48;ESTG_48;71
78 49;ESTG_49;72

```

```

79 # Entidade | Entity: lessons
80 5
81 periodsperweek;teacherids;classids;subjectid
82 4.0;317;268;INF1052_OT
83 4.0;317;267;INF1052_OT
84 4.0;256;270;INF1007_TP
85 4.0;256;270;INF1007_TP
86 4.0;256;270;INF1007_TP
87 # Entidade | Entity: cards
88 32
89 day;subjectid;period;classroomids;teacherids;classids
90 0;INF1052_OT;2;73;317;268
91 0;INF1052_OT;3;73;317;268
92 0;INF1052_OT;4;73;317;268
93 0;INF1052_OT;5;73;317;268
94 0;INF1052_OT;6;73;317;267
95 0;INF1052_OT;7;73;317;267
96 0;INF1052_OT;8;73;317;267
97 0;INF1052_OT;9;73;317;267
98 0;UINF1066_OT;6;51;389;268
99 0;UINF1066_OT;7;51;389;268
100 0;UINF1066_OT;8;51;389;268
101 0;UINF1066_OT;9;51;389;268
102 0;UINF1066_OT;12;51;389;267
103 0;UINF1066_OT;13;51;389;267
104 0;UINF1066_OT;14;51;389;267
105 0;UINF1066_OT;15;51;389;267
106 1;UINF1066_TP;1;51;389;268
107 1;UINF1066_TP;2;51;389;268
108 1;UINF1066_TP;3;51;389;268
109 1;UINF1066_TP;4;51;389;268
110 1;UINF1066_TP;5;51;389;267
111 1;UINF1066_TP;6;51;389;267
112 1;UINF1066_TP;7;51;389;267
113 1;UINF1066_TP;8;51;389;267
114 3;UINF1066_P;2;51;389;268
115 3;UINF1066_P;3;51;389;268
116 3;UINF1066_P;4;51;389;268
117 3;UINF1066_P;5;51;389;268
118 3;UINF1066_P;6;51;389;267
119 3;UINF1066_P;7;51;389;267
120 3;UINF1066_P;8;51;389;267
121 3;UINF1066_P;9;51;389;267

```

1.7.2 XML to SQL

```

1 def ConvertXMLCardsToSQL():
2     entity = 'cards'
3     print("# Entidade | Entity: %s" % entity)
4     es = root.find(entity)
5     print(len(es))
6     sql_all = ""
7     heads = ""
8     for e in es:
9         sql = "INSERT INTO Cards (classids,subjectid,teacherids,classroomids,day,period)
10         VALUES ("
11         i=0
12         heads = ""
13         for k in e.keys():
14             sql = sql + "'%s'," % (e.get(k))
15             if (i==0):
16                 heads = heads + "%s," % k
17             i = i + 1
18         sql = sql.rstrip(',') + ');'
19         #print(sql)
20         sql_all = sql_all + "%s\n" % sql
21         #break;
22     heads = heads.rstrip(',')
23     print(heads)
24     print(sql_all)

```

```

1 # Entidade | Entity: cards
2 1760
3 classids,subjectid,teacherids,classroomids,day,period
4 INSERT INTO Cards (classids,subjectid,teacherids,classroomids,day,period) VALUES ('331','
   CA1048-OT','274','','','3','2');
5 INSERT INTO Cards (classids,subjectid,teacherids,classroomids,day,period) VALUES ('331','
   CA1048-OT','274','','','3','3');
6 INSERT INTO Cards (classids,subjectid,teacherids,classroomids,day,period) VALUES ('331','
   CA1048-OT','274','','','3','4');
7 ...

```

1.7.3 Alunos

```

1 Numero;Ano;DataNascimento;Email;Telemovel;CodigoPostal
2 1012305;1;25-07-1988 08:47;abdulay@gmail.com;969272130;3600-771
3 1012444;1;13-08-1998 10:42;agostinho@gmail.com;946215888;4000-847
4 1012523;1;05-07-1993 23:11;alex@hotmail.com;912916076;4300-251
5 1012163;1;08-09-1988 12:37;alexandre@sal.ipg.pt;966652259;6000-482
6 1700054;1;06-08-1996 17:25;alvaro@hotmail.com;919987690;5400-914
7 1006185;1;14-02-1993 14:33;anabela@sal.ipg.pt;912376753;7400-681
8 1700910;1;06-03-1996 22:13;anderson@gmail.com;969705696;6100-245
9 1012025;1;01-09-1993 02:04;andre@hotmail.com;937342549;6600-687
10 1012022;1;19-01-2000 03:01;andre@hotmail.com;964933667;4600-877
11 1700430;1;13-08-1993 11:40;antonio@hotmail.com;918648427;1000-157
12 1701041;1;19-03-1989 06:52;benjamin@hotmail.com;949347811;7300-416
13 1700094;1;10-05-1988 19:21;bruno@hotmail.com;948513230;9000-722
14 1012026;1;31-12-1991 13:35;bruno@sal.ipg.pt;944440217;4900-163
15 1012225;1;21-01-1994 04:57;bruno@gmail.com;943484082;4200-457
16 1012655;1;02-06-1997 19:21;carla@gmail.com;954743680;6300-825
17 1701062;1;04-08-1997 16:28;carlos@sal.ipg.pt;961426837;8100-501
18 1700637;1;06-12-1990 20:18;cheng@sal.ipg.pt;968067258;2900-593
19 1700104;1;14-01-1996 12:37;daniel@sal.ipg.pt;927367388;7100-661
20 1012208;1;20-07-1989 14:33;dario@gmail.com;922134908;4200-962
21 1011836;1;14-01-1995 07:49;david@sal.ipg.pt;921470704;9400-350
22 1701058;1;06-12-1987 09:45;ezequiel@hotmail.com;934936420;1300-967
23 1701067;1;09-10-1997 18:23;fabio@gmail.com;939240904;5200-409
24 1700616;1;18-02-1994 03:59;fernando@gmail.com;936282609;9400-395
25 ...

```

```

1 students_file_name = "../dados/alunos.txt"
2
3 def ListSudentsFromCSVTextFile(fname):
4     #fname = input("Enter filename: ")
5     infile = open(fname, 'r')
6     for line in infile:
7         line = line.rstrip('\n')
8         print(line)
9
10
11 # ListSudentsFromCSVTextFile(students_file_name)
12
13 def SearchSudentsByNumber(fname, number_to_search):
14     infile = open(fname, 'r')
15     infile.readline() #ignore the 1st line (heads)
16     found = False
17     for line in infile.readlines():
18         line = line.rstrip('\n')
19         data_vector = line.split(';')
20         number = eval(data_vector[1])
21         if (number == number_to_search):
22             found = True
23             break;
24     if found:
25         print("Student number %d found:" % (number_to_search))
26         print(line)
27     else:
28         print("Student number %d not found:" % (number_to_search))
29
30 SearchSudentsByNumber(students_file_name, 1010955)
31
32
33 def SearchSudentsByName(fname, name_to_search):

```

```

34     infile = open(fname, 'r')
35     infile.readline()           #ignore the 1st line (heads)
36     found = False
37     data_vector = []
38     for line in infile.readlines():
39         line = line.rstrip('\n')
40         data_vector = line.split(';')
41         name = data_vector[0]
42         if (name.find(name_to_search) >= 0):
43             found = True
44             break;
45     if found:
46         print("Student name %s found:" % (name_to_search))
47         print(data_vector)
48     else:
49         print("Student name %s not found:" % (name_to_search))
50
51 SearchSudentsByName(students_file_name, "Manuel")

```

Listing 1.2: ListSudentsFromCSVTextFile, SearchSudentsByNumber, SearchSudentsByName

1.7.4 Disciplinas

```

1  Código  Nome  ECTS  Regime  Semestre  Ano
2  UCEE1022  Álgebra e Geometria Analítica 6 S 1 1
3  INF1052  Algoritmos e Estruturas de Dados 6 S 1 1
4  MF1007  Análise Matemática 6 S 1 1
5  MF1008  Introdução à Física 6 S 1 1
6  UINF1066  Introdução à Programação 6 S 1 1
7  UCEE1022  Álgebra e Geometria Analítica 6 S 2 1
8  INF1004  Engenharia de Software I 6 S 2 1
9  MF1008  Introdução à Física 6 S 2 1
10 MF1044  Métodos Numéricos 6 S 2 1
11 UINF1067  Programação 6 S 2 1
12 INF1003  Sistemas Digitais I 6 S 2 1
13 UINF1068  Tecnologias da Internet 6 S 2 1
14 INF1058  Aspetos Sociais da Informática 5 S 1 2
15 INF1056  Bases de Dados I 5 S 1 2
16 INF1017  Controlo Industrial 4,5 S 1 2
17 MF1045  Probabilidades e Estatística 5,5 S 1 2
18 INF1007  Sistemas Digitais II 5 S 1 2
19 INF1035  Sistemas Multimédia 5 S 1 2
20 INF1001  Arquitetura de Computadores 6 S 2 2
21 UINF1069  Programação Avançada 5 S 2 2
22 INF1013  Redes de Computadores 5 S 2 2
23 UINF1070  Robótica 5 S 2 2
24 INF1014  Sistemas Operativos 5 S 2 2
25 INF1060  Computação Gráfica 6 S 1 3
26 INF1012  Engenharia de Software II 6 S 1 3

```

1.7.5 Disciplinas alunos

```

1  CdDisciplina  Cdaluno  Nota  DataAprovacao
2  UCEE1022  1012305
3  INF1052  1012305
4  MF1007  1012305
5  MF1008  1012305
6  UINF1066  1012305
7  UCEE1022  1012305
8  INF1004  1012305
9  MF1008  1012305
10 MF1044  1012305
11 UINF1067  1012305
12 INF1003  1012305
13 UINF1068  1012305
14 UCEE1022  1012444
15 INF1052  1012444
16 MF1007  1012444
17 MF1008  1012444
18 UINF1066  1012444

```

19	UCEE1022	1012444
20	INF1004	1012444
21	...	