



**Universidade do Minho**

UNIVERSIDADE DO MINHO

LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

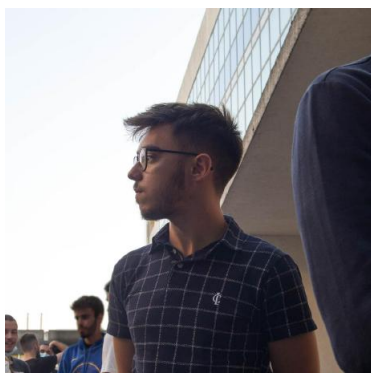
Gestor de Distribuição de Turnos TP e PL

Simão Quintela  
(A97444)

Hugo Rocha  
(A96463)

Miguel Freitas  
(A91635)

2 de junho de 2023



# Índice

<b>1</b>	<b>Agradecimentos</b>	<b>3</b>
<b>2</b>	<b>Introdução</b>	<b>4</b>
<b>3</b>	<b>Geração de distribuições</b>	<b>5</b>
3.1	Construção da matriz S . . . . .	5
3.2	Inicialização das matrizes no Solver . . . . .	5
3.2.1	Matriz A . . . . .	5
3.2.2	Matriz P . . . . .	6
3.2.3	Matriz O . . . . .	6
3.3	Restrições do solver . . . . .	6
3.4	Minimização de sobreposições . . . . .	8
3.5	Funções de análise à distribuição . . . . .	9
<b>4</b>	<b>Visualização de Horários</b>	<b>11</b>
4.1	Página de upload de ficheiros e geração de distribuições . . . . .	11
4.2	Página de visualização de horários . . . . .	12
4.3	Trocas entre turnos . . . . .	13
4.4	Exportar horários . . . . .	14
<b>5</b>	<b>Conclusão</b>	<b>15</b>

# Capítulo 1

## Agradecimentos

Agradecemos ao NECC - Núcleo de Estudantes de Ciências da Computação, pelo apoio no desenvolvimento e continuidade do projeto, que irá aumentar consideravelmente a facilidade da gestão de turnos feita por professores e alunos.

Agradecemos também, com especial apreço, ao professor António José Borba Ramires Fernandes pela disponibilidade que demonstrou, transmitindo conhecimento e ideias para o bom funcionamento e desenvolvimento do projeto.

## Capítulo 2

# Introdução

Neste projeto foi-nos proposta a implementação de um gestor de distribuição de turnos com vista a distribuir todos os alunos do curso pelos turnos das respetivas unidades curriculares nas quais estão inscritos.

Este projeto surgiu dada a necessidade de encontrar uma solução (quase) óptima para a distribuição de alunos pelos respetivos turnos nas diferentes unidades curriculares. Na distribuição gerada, procurou-se minimizar sobreposições de aulas para cada aluno, balancear os turnos e evitar ultrapassar a lotação das salas de aula.

Para além da distribuição, foi também proposto o desenvolvimento de duas interfaces. A primeira deveria permitir a realização de trocas entre alunos de forma simples e prática, enquanto que o propósito da segunda seria mostrar informação sobre o estado da distribuição, contando com eventuais manipulações feitas pelo utilizador.

Nas próximas páginas estarão descritas as estratégias utilizadas para resolver os problemas que surgiram durante o desenvolvimento deste programa.

## Capítulo 3

# Geração de distribuições

Na apresentação deste projeto tornou-se claro que o problema que estava perante nós era um **problema de alocação**.

Aproveitando o conteúdo apresentado na uc **Lógica Computacional**, decidimos utilizar um **CP-SAT solver** para modelar o problema, usando fórmulas booleanas e programação inteira. O solver escolhido foi o **Cp\_Model** da biblioteca **or-tools** desenvolvida pela Google.

### 3.1 Construção da matriz S

Para melhor comodismo, decidimos criar uma matriz S que guarda toda a informação acerca dos horários de todas as aulas do curso. Nesse sentido, lemos um ficheiro csv com essa informação e criámos a matriz S de 6 dimensões com o seguinte formato:

$$\forall_{year} \cdot \forall_{semester} \cdot \forall_{uc} \cdot \forall_{typeclass} \cdot \forall_{shift} \cdot \forall_{slot} \cdot S_{year, semester, uc, typeclass, shift, slot} = 1$$

se e só se a unidade curricular **uc** pertencer ao ano **year** e ao semestre **semester** e existir uma aula do tipo **typeclass**, do turno **shift** no slot **slot**.

Os slots são tuplos com duas componentes onde a primeira componente é um inteiro de 1 a 5 que faz referência ao dia da semana pela respetiva ordem. A segunda componente é também um tuplo de duas componentes onde a primeira indica uma hora e a segunda os minutos. Foram considerados slots de 30 minutos para contemplar todas as possibilidades de datas de início e fim de aulas.

### 3.2 Inicialização das matrizes no Solver

Para começar, optamos por inicializar matrizes com variáveis do solver, com vista a fazer a distribuição ótima dos alunos pelos respetivos turnos. Para isso, utilizamos três matrizes multidimensionais, as matrizes **A**, **P** e **O**.

#### 3.2.1 Matriz A

A matriz A possui 7 dimensões e é responsável por garantir a alocação dos alunos a determinadas aulas. A matriz A tem o seguinte formato:

$$\forall_{studentNr} \cdot \forall_{year} \cdot \forall_{semester} \cdot \forall_{uc} \cdot \forall_{typeclass} \cdot \forall_{shift} \cdot \forall_{slot} \cdot A_{studentNr, year, semester, uc, typeclass, shift, slot} = 1$$

se e só se o aluno **student** estiver alocado a uma aula da unidade curricular **uc**, do tipo **typeclass**, do turno **shift**, no slot **slot** onde a unidade curricular pertence ao ano **year** e ao semestre **semester**. Em caso contrário, o valor é **0**.

### 3.2.2 Matriz P

A matriz P possui 6 dimensões e é responsável por garantir a alocação dos alunos aos turnos das respectivas unidades curriculares. A matriz P tem o seguinte formato:

$$\forall_{studentNr} \cdot \forall_{year} \cdot \forall_{semester} \cdot \forall_{uc} \cdot \forall_{typeclass} \cdot \forall_{shift} \cdot \\ P_{studentNr, year, semester, uc, typeclass, shift} = 1$$

se e só se o aluno **student** estiver alocado ao turno **shift** de uma aula do tipo **typeclass** da unidade curricular **uc**, onde a unidade curricular pertence ao ano **year** e ao semestre **semester**. Caso o contrário, o valor é **0**.

### 3.2.3 Matriz O

A matriz O apenas possui 2 dimensões e é responsável por controlar o número de **sobreposições** de cada aluno em cada slot. A matriz O tem o seguinte formato:

$$\forall_{studentNr} \cdot \forall_{slot} \cdot O_{studentNr, slot} = abs(N - 1)$$

, onde N é o número de aulas do aluno student no slot slot.

## 3.3 Restrições do solver

- Um aluno só pode estar alocado a uma aula num determinado slot se essa aula existir nesse slot.

$$\forall_{studentNr} \cdot \forall_{year} \cdot \forall_{uc} \cdot \forall_{typeclass} \cdot \forall_{shift} \cdot \forall_{slot} \cdot \\ A_{studentNr, year, semester, uc, typeclass, shift, slot} \leq S_{year, semester, uc, typeclass, shift, slot}$$

- Se um aluno estiver alocado a uma aula de um determinado turno então ele está alocado a esse turno.

$$\forall_{studentNr} \cdot \forall_{year} \cdot \forall_{uc} \cdot \forall_{typeclass} \cdot \forall_{shift} \cdot \forall_{slot} \cdot \\ P_{student, year, semester, uc, typeclass, shift} == A_{student, year, semester, uc, typeclass, shift, slot}$$

- Um aluno só pode estar alocado a um turno para cada tipo de aula de uma dada UC.

$$\forall_{studentNr} \cdot \forall_{year} \cdot \forall_{uc} \cdot \forall_{typeclass} \cdot \\ \sum_{shift < shifts} P_{student, year, semester, uc, typeclass, shift} == 1$$

- Para cada turno, existe um número mínimo de alunos (  $(N / K) - (N \times 0.1)$  ) onde:

N = número de alunos inscritos na UC

K = número de turnos dessa UC

$$\forall_{year} \cdot \forall_{uc} \cdot \forall_{typeclass} \cdot \forall_{shift}$$

$$\sum_{student} P_{student, year, semester, uc, typeclass, shift} \leq N/K - (N * 0.1)$$

- O número de alunos alocados a uma dada aula tem de ser igual ou inferior à capacidade da sala (nesta restrição assumimos uma taxa de presença nas aulas de 70%)

$$\forall_{slot} \cdot \forall_{year} \cdot \forall_{uc} \cdot \forall_{typeclass} \cdot \forall_{shift}$$

$$\sum_{student} A_{student, year, semester, uc, typeclass, shift, slot} \leq room\_capacity/0.7$$

- O valor de **O[student][slot]** é igual ao valor absoluto do número de aulas do aluno student naquele slot (N) menos um (preenchimento da matriz O do solver).

$$\forall_{student} \cdot \forall_{slot} \cdot \\ O_{student, slot} == |N - 1|$$

Onde N toma o valor de:

$$\forall_{year} \cdot \forall_{uc} \cdot \forall_{typeclass} \cdot \forall_{shift}$$

$$\sum_{student} A_{student, year, semester, uc, typeclass, shift, slot}$$

- Os alunos com unidades curriculares de apenas um ano não podem ter sobreposições de aulas.

$$\forall_{student} \cdot \forall_{slot} \cdot \\ N \leq 1$$

Onde N (número de aulas no slot acima) toma o valor de:

$$\forall_{year} \cdot \forall_{uc} \cdot \forall_{typeclass} \cdot \forall_{shift}$$

$$\sum_{student} A_{student, year, semester, uc, typeclass, shift, slot}$$

- No pior dos casos existe uma sobreposição de duas aulas num slot.

$$\forall_{student} \cdot \forall_{slot} \cdot \\ O_{student, slot} \leq 1$$

### 3.4 Minimização de sobreposições

Para minimizar as sobreposições, basta-nos minimizar o valor de cada campo da matriz  $O$ . Visto que  $O[\text{student}][\text{slot}] = \text{abs}(N-1)$  com  $N = \text{número de aulas no slot } \mathbf{slot}$ , o menor valor que  $\text{abs}(N-1)$  pode tomar é 0, com  $N = 1$ , logo estamos a forçar ao máximo que haja apenas uma aula em cada slot. Quanto maior o valor de  $N$ , maior o valor de  $\text{abs}(N-1)$  logo minimizar o valor de  $\text{abs}(N-1)$  implica minimizar o valor de  $N$  exceto no caso  $N = 0$  e  $N = 1$  que é o caso ótimo.



### 3.5 Funções de análise à distribuição

Foi também um criado um menu para diferentes análises da distribuição que foi gerada.

```
Choose an option:

1 - Students overlaps and number of conflicts.
2 - Shifts distribution.
3 - Overlaps per shift.
4 - Students allocated number per UC.
9 - Leave Menu.

Option:
```

Figura 3.1: Menu de análise à distribuição

Entre essas análises, temos:

- Distribuição dos alunos pelos turnos de todas as unidades curriculares

```
{'Análise': {'TP1': 44, 'TP2': 67},
'Autômatos e Linguagens Formais': {'T1': 81, 'TP1': 81},
'Computação Gráfica': {'PL1': 14, 'PL2': 34, 'PL5': 14, 'T1': 62},
'Cálculo de Programas': {'T1': 123, 'TP1': 51, 'TP2': 72},
'Geometria': {'T1': 107, 'TP1': 43, 'TP2': 64},
'Interação e Concorrência': {'T1': 73, 'TP1': 73},
'Laboratório de Algoritmia I': {'PL1': 34, 'PL2': 47, 'PL3': 25},
'Laboratório de Algoritmia II': {'PL1': 42, 'PL2': 29},
'Matemática Discreta': {'T1': 121, 'TP1': 48, 'TP2': 73},
'Programação Concorrente': {'PL1': 30, 'PL2': 44, 'T1': 74},
'Programação Imperativa': {'T1': 183, 'TP1': 70, 'TP2': 63, 'TP3': 50},
'Programação Orientada aos Objetos': {'PL1': 52, 'PL2': 35, 'T1': 87},
'Semântica das Linguagens de Programação': {'T1': 69, 'TP1': 69},
'Sistemas Operativos': {'PL1': 37, 'PL2': 25, 'PL3': 43, 'T1': 105},
'Sistemas de Computação': {'PL1': 55,
                           'PL2': 25,
                           'PL3': 31,
                           'PL4': 46,
                           'T1': 157},
'Teoria de Números Computacional': {'T1': 50, 'TP1': 22, 'TP2': 28},
'Álgebra Universal e Categorias': {'T1': 147, 'TP1': 68, 'TP2': 44, 'TP3': 35}}
```

Figura 3.2: Distribuição dos alunos pelos turnos

- Sobreposições para todos os alunos e número total de alunos com conflitos

```
74654: [],
74805: [],
75365: ['Terça / 16:00 - 17:00 / Programação Imperativa / T1',
       'Terça / 16:00 - 18:00 / Cálculo de Programas / TP2',
       'Quarta / 09:00 - 10:00 / Sistemas de Computação / T1',
       'Quarta / 09:00 - 11:00 / Programação Orientada aos Objetos / T1'],
76101: [],
77693: [],
78500: [],
79160: ['Terça / 16:00 - 17:00 / Programação Imperativa / T1',
       'Terça / 16:00 - 18:00 / Cálculo de Programas / TP2',
       'Quarta / 09:00 - 10:00 / Sistemas de Computação / T1',
       'Quarta / 09:00 - 11:00 / Programação Orientada aos Objetos / T1'],
79613: [],
80112: ['Quarta / 09:00 - 10:00 / Sistemas de Computação / T1',
       'Quarta / 09:00 - 11:00 / Programação Orientada aos Objetos / T1'],
80281: ['Quinta / 11:00 - 13:00 / Autômatos e Linguagens Formais / T1',
       'Quinta / 11:00 - 13:00 / Computação Gráfica / PL2'],
```

Figura 3.3: Sobreposições para todos os alunos

```
124 de 291 alunos (42.61%) possuem conflitos no seu horário.
```

Figura 3.4: Número de alunos com conflitos

- Número de alocados em cada unidade curricular

```
{'Análise': 111,
 'Autômatos e Linguagens Formais': 81,
 'Computação Gráfica': 62,
 'Cálculo de Programas': 123,
 'Geometria': 107,
 'Interação e Concorrência': 73,
 'Laboratório de Algoritmia I': 106,
 'Laboratório de Algoritmia II': 71,
 'Matemática Discreta': 121,
 'Programação Concorrente': 74,
 'Programação Imperativa': 183,
 'Programação Orientada aos Objetos': 87,
 'Semântica das Linguagens de Programação': 69,
 'Sistemas Operativos': 105,
 'Sistemas de Computação': 157,
 'Teoria de Números Computacional': 50,
 'Álgebra Universal e Categorias': 147}
```

Figura 3.5: Número de alunos alocados em cada unidade curricular

- Turnos com mais sobreposições para todas as unidades curriculares e respectivos alunos

```
{'Análise': {'TP1': 1, 'TP2': 2},
 'Autômatos e Linguagens Formais': {'T1': 23, 'TP1': 2},
 'Computação Gráfica': {'PL1': 11, 'PL2': 9, 'PL5': 0, 'T1': 0},
 'Cálculo de Programas': {'T1': 0, 'TP1': 9, 'TP2': 35},
 'Geometria': {'T1': 5, 'TP1': 0, 'TP2': 1},
 'Interação e Concorrência': {'T1': 1, 'TP1': 16},
 'Laboratório de Algoritmia I': {'PL1': 1, 'PL2': 0, 'PL3': 1},
 'Laboratório de Algoritmia II': {'PL1': 0, 'PL2': 0},
 'Matemática Discreta': {'T1': 16, 'TP1': 0, 'TP2': 1},
 'Programação Concorrente': {'PL1': 0, 'PL2': 0, 'T1': 0},
 'Programação Imperativa': {'T1': 85, 'TP1': 40, 'TP2': 0, 'TP3': 19},
 'Programação Orientada aos Objetos': {'PL1': 0, 'PL2': 0, 'T1': 61},
 'Semântica das Linguagens de Programação': {'T1': 0, 'TP1': 14},
 'Sistemas Operativos': {'PL1': 0, 'PL2': 0, 'PL3': 0, 'T1': 0},
 'Sistemas de Computação': {'PL1': 0, 'PL2': 1, 'PL3': 0, 'PL4': 0, 'T1': 45},
 'Teoria de Números Computacional': {'T1': 15, 'TP1': 0, 'TP2': 0},
 'Álgebra Universal e Categorias': {'T1': 70, 'TP1': 39, 'TP2': 7, 'TP3': 0}}
```

Figura 3.6: Sobreposições em cada turno

```
{'Computação Gráfica': {'PL1': ['A99685',
 'A92565',
 'A90547',
 'A67312',
 'A96357',
 'A90167',
 'A99506',
 'A94561'],
 'PL2': ['A70462',
 'A80281',
 'A93454',
 'A93364',
 'A95661',
 'A94861'],
 'PL5': ['A96667',
 'A92795',
 'A99761',
 'A92361',
 'A97361'],
 'T1': []}}
```

Figura 3.7: Alunos com sobreposições a uma determinada UC

## Capítulo 4

# Visualização de Horários

Para o utilizador se sentir confortável na utilização do programa foi criada uma web page que corre localmente na máquina do utilizador. Além da visualização, também é oferecida a possibilidade gerar distribuições a partir da página, realizar o upload de ficheiros necessários, trocar alunos de turno, exportar os horários de todos os alunos e exportar o horário dos 3 anos.

### 4.1 Página de upload de ficheiros e geração de distribuições

**Upload Files**

Files should have the .csv extension and follow the specified format in the user guide.

**Schedule**

Insert a file Schedule  
Only accepts .CSV files

**Students**

Insert a file Students  
Only accepts .CSV files

**Rooms**

Insert a file Rooms  
Only accepts .CSV files

Upload

Generate

Figura 4.1: Página de upload dos ficheiros CSV

## 4.2 Página de visualização de horários

Na página de visualização de horários, o utilizador insere o número do aluno que quer visualizar e obtém o horário do mesmo. O horário segue um padrão de cores para que seja imediata a deteção de problemas no mesmo. O padrão é o seguinte:

- Ucs de 1<sup>o</sup>, 2<sup>o</sup> e 3<sup>o</sup> ano estão a azul, vermelho e verde, respetivamente.
- Ucs que estão em sobreposição estão a cinzento.
- Aulas práticas e laboratoriais estão com uma ligeira variação do tom de cor do respetivo ano.

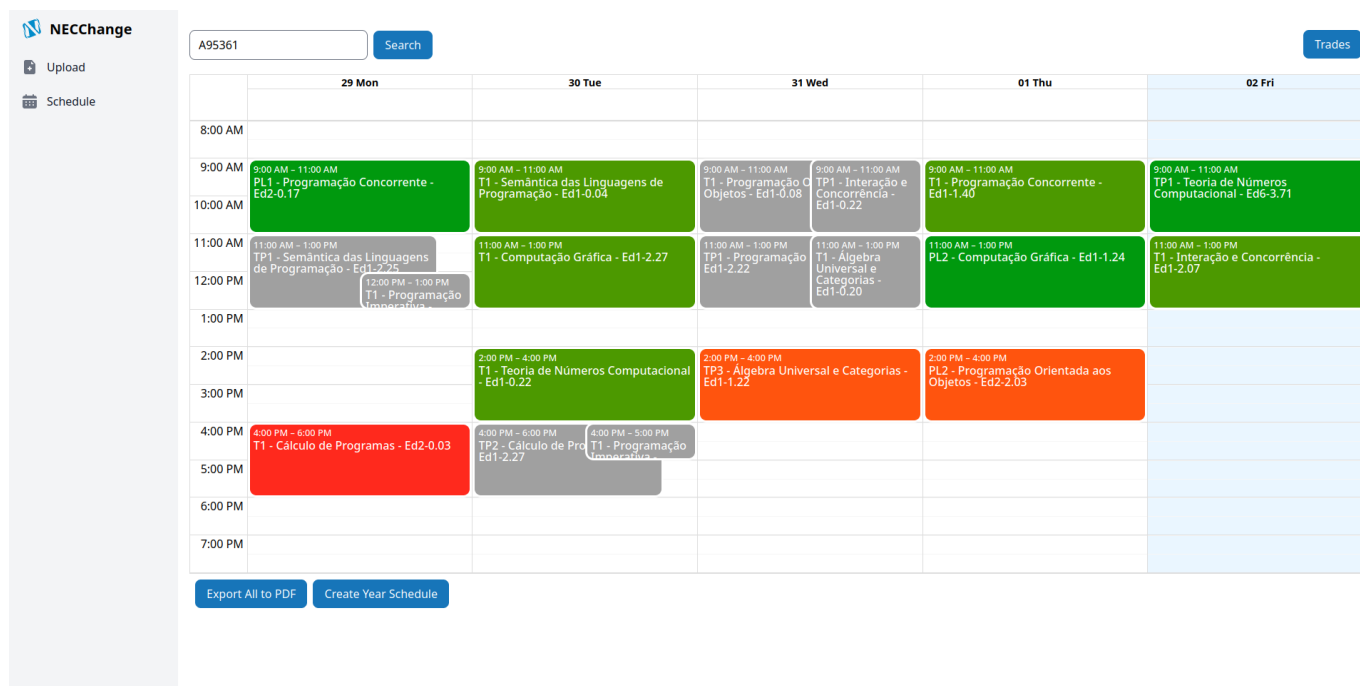


Figura 4.2: Página de visualização dos horários para cada aluno

### 4.3 Trocas entre turnos

Para as trocas de turnos foi implementado um menu que permite ver todas as aulas práticas e laboratoriais onde o utilizador pode seleccionar o turno para o qual deseja mudar.

**Trades**

- Laboratório de Algoritmia II - PL2
- Autómatos e Linguagens Formais - TP1
- Sistemas Operativos - PL2
- Computação Gráfica - PL1
- Teoria de Números Computacional - TP1
- Programação Concorrente - PL1
- Interação e Concorrência - TP1
- Semântica das Linguagens de Programação - TP1

**Submit**

Figura 4.3: Menu de trocas

## 4.4 Exportar horários

Implementamos também uma forma de exportar os horários de todos os alunos. O horário segue um padrão de cores semelhante à utilizada no web site.

A95361

	Segunda	Terça	Quarta	Quinta	Sexta
8h-9h					
9h-10h	TNC TP2 (Ed6-3.71)	SLP T1 (Ed1-0.04)	POO T1 (Ed1-0.08)/IC TP1 (Ed1-0.22)	PC T1 (Ed1-1.40)	PC PL2 (Ed1-2.19)
10h-11h	TNC TP2 (Ed6-3.71)	SLP T1 (Ed1-0.04)	POO T1 (Ed1-0.08)/IC TP1 (Ed1-0.22)	PC T1 (Ed1-1.40)	PC PL2 (Ed1-2.19)
11h-12h	SLP TP1 (Ed1-2.25)	CG T1 (Ed1-2.27)	PI TP1 (Ed1-2.22)/AUC T1 (Ed1-0.20)	CG PL2 (Ed1-1.24)	IC T1 (Ed1-2.07)
12h-13h	PI T1 (Ed1-0.20)/SLP TP1 (Ed1-2.25)	CG T1 (Ed1-2.27)	PI TP1 (Ed1-2.22)/AUC T1 (Ed1-0.20)	CG PL2 (Ed1-1.24)	IC T1 (Ed1-2.07)
13h-14h					
14h-15h		TNC T1 (Ed1-0.22)	AUC TP3 (Ed1-1.22)	POO PL2 (Ed2-2.03)	
15h-16h		TNC T1 (Ed1-0.22)	AUC TP3 (Ed1-1.22)	POO PL2 (Ed2-2.03)	
16h-17h	CP T1 (Ed2-0.03)	PI T1 (Ed2-0.03)/CP TP2 (Ed1-2.27)			
17h-18h	CP T1 (Ed2-0.03)	CP TP2 (Ed1-2.27)			
18h-19h					
19h-20h					

Figura 4.4: Exemplo de um horário exportado

## Capítulo 5

# Conclusão

Dado por terminado o projeto, concluimos que o mesmo fez jus aos ensinamentos adquiridos ao longo da licenciatura visto que foram usadas ferramentas apresentadas em unidades curriculares transatas. Sentimos que o curso nos providenciou todos os conhecimentos necessários para o bom desenvolvimento do projeto, principalmente, neste caso, para a utilização dos solvers e de toda a lógica por trás das condições implementadas.

Para além disso, dada a dimensão do projeto foi necessário um grande planeamento do mesmo e consideramos que a boa planificação foi uma peça chave para o sucesso do mesmo.

Esperemos que este projeto tenha atendido às necessidades do orientador, António Ramires Fernandes, e que o mesmo seja útil para a direção de curso de forma a permitir automatizar a distribuição de alunos pelos diferentes turnos.