



Universidade do Minho

UNIVERSIDADE DO MINHO

LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

Guião de Utilização e Relatório
NECChange - Gestor de Distribuição de Turnos

Carlos Daniel Silva Fernandes
(A102499)

José Bernardo Moniz Fernandes
(A102497)

Pedro Augusto Camargo
(A102504)

Renato Pereira Garcia
(A101987)

30 de maio de 2025

Conteúdo

1	Agradecimentos	3
2	Introdução	4
3	Setup do programa	5
3.1	Windows	5
3.2	Linux	6
4	Execução do programa	7
4.1	Windows e Linux	7
5	Carregamento de ficheiros	8
6	Visualização dos horários	9
7	Troca de turnos	11
8	Visualização dos horários dos diversos anos	12
9	Análise de Dados	14
10	Manutenção do Solver e Possíveis novas restrições	16
10.1	Construção da matriz S	16
10.2	Inicialização das matrizes no Solver	16
10.2.1	Matriz A	16
10.2.2	Matriz P	17
10.2.3	Matriz O	17
10.3	Restrições do solver	17
11	Conclusão	19

Capítulo 1

Agradecimentos

Agradecemos ao Simão Quintela, membro do NECC - Núcleo de Estudantes de Ciências da Computação, por toda a disponibilidade em ajudar na melhor compreensão do que já tinha sido implementado do projeto.

Agradecemos também aos professores Fernando Miranda e António Ramires pela ajuda e acompanhamento ao longo do projeto.

Capítulo 2

Introdução

Neste projeto foi-nos proposta a melhoria de uma interface com vários recursos para os utilizadores 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, bem como um upgrade na possibilidade de aumentar/diminuir dinamicamente as restrições dessa mesma distribuição.

Foram acrescentadas novas interfaces ao projeto, como é o caso de uma página de análise de dados, acompanhado pela melhoria das funcionalidades das já existentes.

Capítulo 3

Setup do programa

3.1 Windows

1- Instalar o Anaconda a partir do site

Depois de terminada a instalação do anaconda, deve-se pesquisar pelo **Anaconda Prompt** no computador pessoal. Todos os passos seguintes devem ser feitos neste mesmo sítio.

2 - Ainda antes de se utilizar o anaconda prompt, deve-se transferir o ficheiro `logica_env.yml`, presente na mesma pasta que este guião de utilização.

3 - Agora no anaconda prompt, coloca-se o seguinte comando:

`conda env create -n necc -f logica_env.yml`

(deve-se garantir que o ficheiro `.yml` está presente na mesma diretoria do anaconda onde estamos a executar o comando)

4 - Finalizado o anterior comando, executa-se o comando: **`conda activate necc`**

5 - Executar o comando **`conda install anaconda::git`** já dentro do ambiente **`necc`**, bem como o comando **`conda install conda-forge::nodejs`**

6 - Dar clone do repositório do github, com o seguinte comando:

`git clone https://github.com/RenatoP18/NECChange-Solver.git`

7 - Como o windows apenas reconhece o python e não python3, deve-se fazer os seguintes comandos:

- **`cd anaconda3`**

- **`cd envs`**

- **`cd necc`**

- **`notepad python3.bat`**

O anterior comando irá abrir um editor de texto, onde se deverá escrever o seguinte:

`@python %*`

No fim, deve-se salvar esse ficheiro e fechar. Para confirmar a criação, volta-se ao anaconda prompt e escrever **`dir`**. Caso o ficheiro exista, ele aparecerá lá e existe a certeza que este está criado. Por fim, deve-se

voltar à diretoria inicial antes de se efetuarem estes comandos, ou seja, à diretoria que aparece estritamente antes da diretoria **anaconda3**. Isso é feito recorrendo ao comando **cd ..**, as vezes que forem necessárias.

8 - Em conclusão, efetua-se os dois comandos seguintes:

- **conda install pandas=2.2.2**

- **pip install ortools==9.11.4210**

3.2 Linux

Em linux, o setup do programa é efetuado exatamente da mesma forma. O ponto 7 poderá não ser necessário efetuar.

Capítulo 4

Execução do programa

Um dos objetivos deste projeto era a execução do programa via front-end, onde é feito o upload de ficheiros, geração de horários, visualização, alteração de horários, estatísticas de alocação, estatísticas de sobreposição.

4.1 Windows e Linux

Para a execução do programa, é apenas necessário fazer agora o seguinte (para ir para a diretoria onde se vai executar o programa):

- **cd NECChange-Solver**
- **cd web**

Com estes dois comandos, é agora possível efetuar os seguintes comandos, visto que o utilizador já está na diretoria web:

- **npm i -force**
- **npm run dev**

Com isso, o utilizador está agora pronto a navegar na interface do programa, em <http://localhost:3000>, endereço que aparece no anaconda prompt.

Nota: Os passos anteriores, tanto em **Windows** como em **Linux**, apenas são necessários na primeira vez. Das vezes seguintes em que se utiliza o programa, apenas é necessário ativar o ambiente e seguir até à diretoria web, executando depois o comando **npm run dev**.

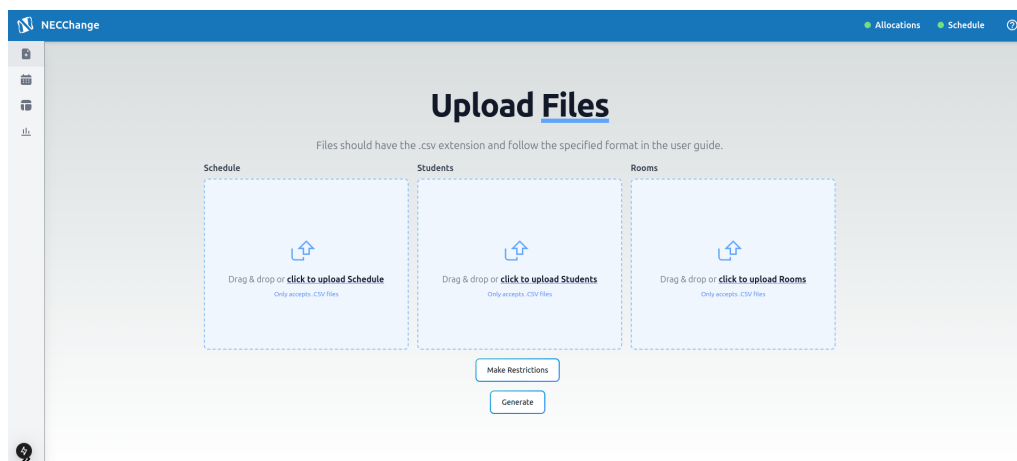
Capítulo 5

Carregamento de ficheiros

Já na página do servidor local, encontra-se a secção de **Upload**, como página principal. Aqui, devem ser carregados três ficheiros no formato **.csv**.

- O primeiro ficheiro deve conter a informação do horário de cada Unidade Curricular (UC) do curso. O seu nome deverá ser **horario.csv**.
- O segundo ficheiro deve conter a informação sobre a inscrição dos alunos nas respetivas UCs. O seu nome deverá ser **inscritos_anon.csv**.
- O terceiro ficheiro deve conter informação sobre a capacidade de cada sala. O seu nome deverá ser **salas.csv**.

O utilizado pode ainda pedir, antes de gerar os horários, mudança nas restrições à sua escolha. Por fim, o utilizador pode solicitar a geração do novo horário ao clicar no botão **Generate**. Este botão inicia a execução do algoritmo que processa os dados carregados e gera o novo horário. Similarmente ao processo de upload, o utilizador é informado do sucesso ou falha da operação.

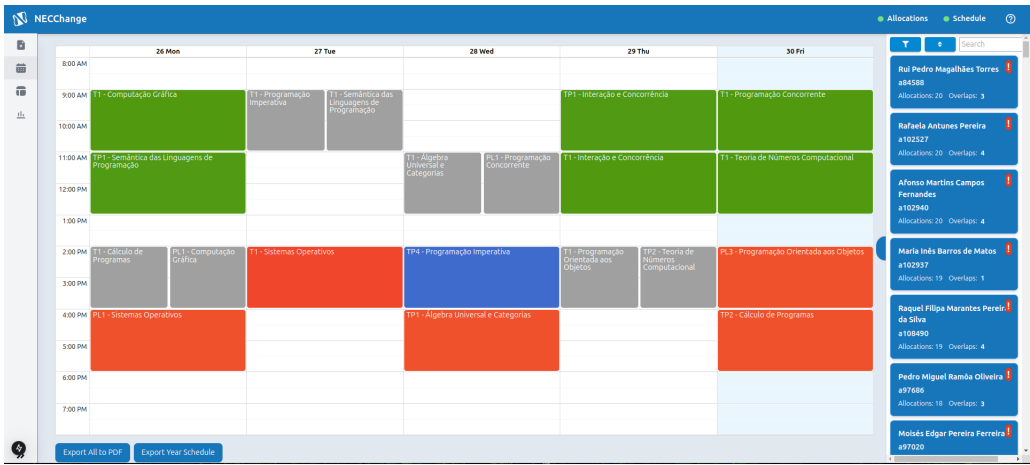


Visualização da página principal, onde são gerados os horários

Capítulo 6

Visualização dos horários

Para visualizar os horários gerados, clica-se no separador **Schedule** (segundo elemento na sidebar), do lado esquerdo, onde podemos consultar o horário gerado para cada aluno, colocando o respetivo número mecanográfico, do lado direito (para aparecer o horário gerado do aluno, deve-se carregar em cima). Essa pesquisa pode ser filtrada também por alocações ou overlaps, bem como ordenada pelos mesmos critérios. Para melhor compreensão do horário, as cadeiras de diferentes anos possuem diferentes cores, nomeadamente, às unidades curriculares do primeiro ano ficou associada a cor azul, às de segundo a cor vermelha e, por fim, as disciplinas do último ano ficaram com a cor verde. As sobreposições aparecem em cor cinzenta. Para além disso, é possível também exportar os horários dos alunos em formato pdf. Estes passam a estar disponíveis na pasta Data, dentro do projeto.



Visualização de um horário

A81663

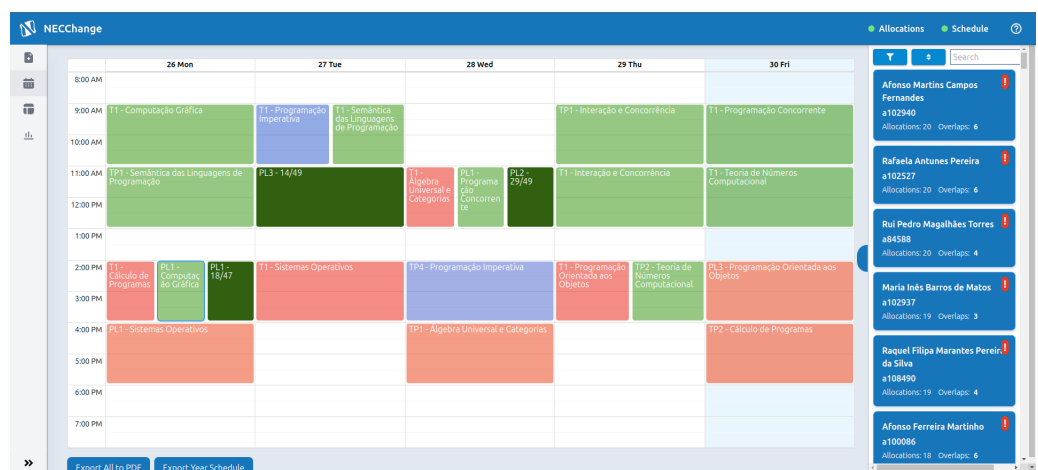
	Segunda	Terça	Quarta	Quinta	Sexta
8h-9h					
9h-10h	AN PL1 (Ed6-3.72)/CG T1 (Ed1-1.25)	PI T1 (Ed1-0.08)		IC TP1 (Ed1-1.40)	
10h-11h	AN PL1 (Ed6-3.72)/CG T1 (Ed1-1.25)	PI T1 (Ed1-0.08)		IC TP1 (Ed1-1.40)	
11h-12h		CG PL3 (Ed1-2.22)	AUC T1 (Ed1-0.04)	IC T1 (Ed1-1.40)	
12h-13h		CG PL3 (Ed1-2.22)	AUC T1 (Ed1-0.04)	IC T1 (Ed1-1.40)	
13h-14h					
14h-15h	CP T1 (Ed1-0.04)	SO T1 (Ed1-0.08)	PI TP4 (Ed1-1.20)	POO T1 (Ed1-0.08)	POO PL1 (Ed7-0.07)
15h-16h	CP T1 (Ed1-0.04)	AN TP2 (Ed6-3.72)/SO T1 (Ed1-0.08)	PI TP4 (Ed1-1.20)	POO T1 (Ed1-0.08)	POO PL1 (Ed7-0.07)
16h-17h	SO PL1 (Ed1-2.19)	AN TP2 (Ed6-3.72)	AUC TP1 (Ed1-1.05)		CP TP2 (Ed2-1.03)
17h-18h	SO PL1 (Ed1-2.19)	AN TP2 (Ed6-3.72)	AUC TP1 (Ed1-1.05)		CP TP2 (Ed2-1.03)
18h-19h					
19h-20h					

Visualização de um horário de um aluno depois de exportado

Capítulo 7

Troca de turnos

A troca de turnos para um dado aluno deve ser feita também na mesma secção **Schedule**. Para um dado aluno, dentro do seu horário, ao clicarmos uma/duas vezes numa slot de uma determinada UC, é possível obter todas as outras possibilidades para onde podemos transferir esse aluno. No caso abaixo ilustrado, está-se a tentar efetuar uma troca de turno da Computação Gráfica. Como tal aparecem as opções, com a mesma cor, para onde é possível efetuar essa troca.

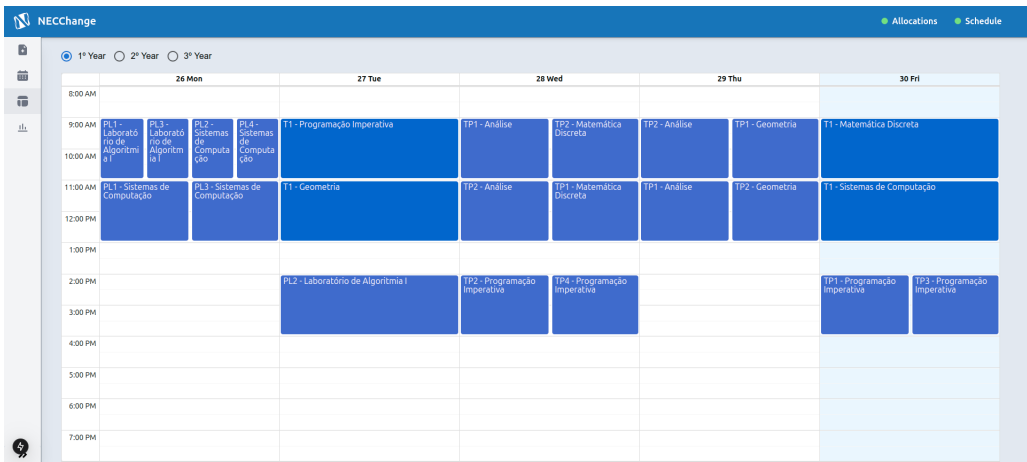


Demonstração de como aparece o horário aquando de uma mudança

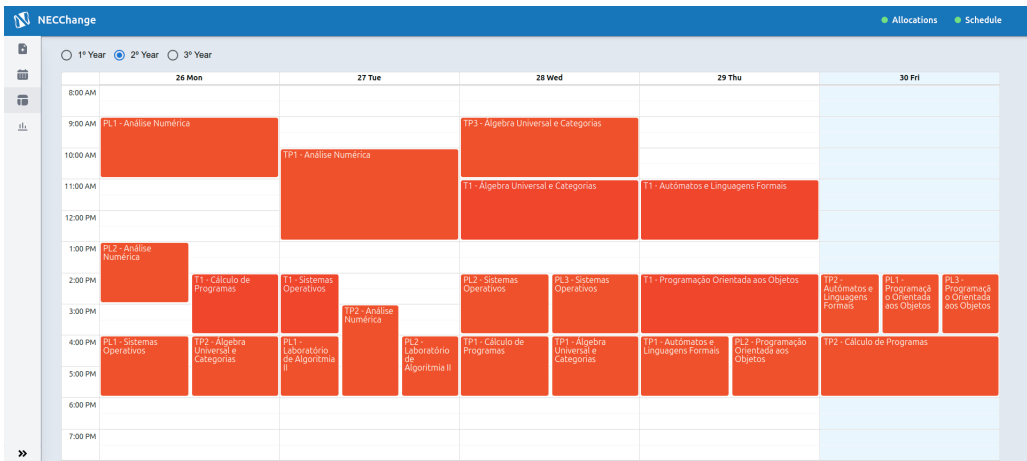
Capítulo 8

Visualização dos horários dos diversos anos

Na terceira aba da sidebar, é possível observar os horários completos dos diversos anos.



Horário do 1ºano

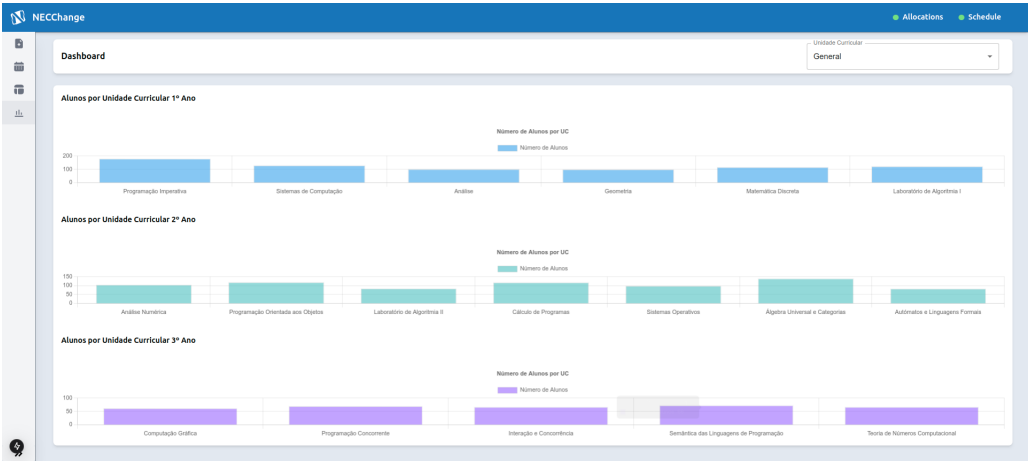


Horário do 2ºano

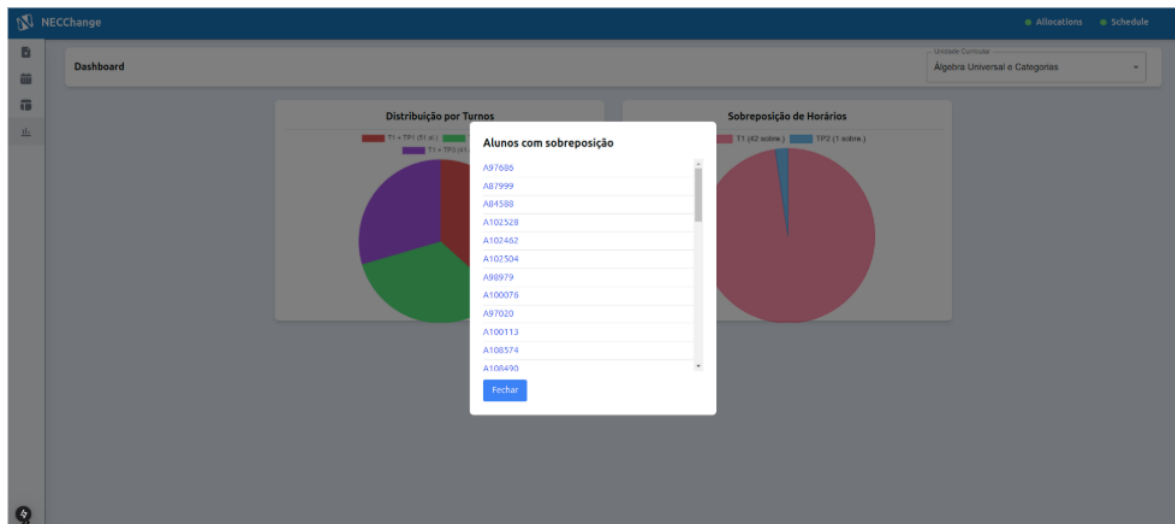
Capítulo 9

Análise de Dados

Para finalizar na sidebar, foi criada uma nova página de análise de dados. Esta página é responsável por mostrar a distribuição dos alunos por unidade curricular dos diferentes anos. Para além disso, é possível ainda alternar entre as diferentes UC's para ver, em particular, a distribuição por turnos e as sobreposições que existem. Nas sobreposições, é possível ver o número dos alunos que têm essa sobreposição, sendo ainda disponibilizada a possibilidade de carregar no número do aluno e ser enviado para a página dos horários dos alunos.



Página com distribuição por unidades curriculares



Página com distribuição dos alunos por turnos e as sobreposições, dada uma UC

Capítulo 10

Manutenção do Solver e Possíveis novas restrições

O projeto continuou a utilizar um CP-SAT solver para modelar o problema de alocação, usando fórmulas booleanas e programação inteira. O solver escolhido foi o Cp Model da biblioteca or-tools desenvolvida pela Google.

10.1 Construção da matriz S

Guarda toda a informação acerca dos horários de todas as aulas do curso. Nesse sentido, lê-se um ficheiro CSV com essa informação e cria-se a matriz S de 6 dimensões com o seguinte formato:

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

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: uma 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.

10.2 Inicialização das matrizes no Solver

Matrizes **A**, **P** e **O**.

10.2.1 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:

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

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.

10.2.2 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:

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

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.

10.2.3 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:

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

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

10.3 Restrições do solver

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

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

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

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

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

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

Para cada turno, existe um número mínimo de alunos

$$\left(\frac{N}{K} \right) - (N \times 0.1)$$

onde:

- *N* = número de alunos inscritos na UC
- *K* = número de turnos dessa UC

Para além de todas as restrições já enunciadas, temos ainda as seguintes, explicadas de forma breve:

- 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%).

- O valor de $O[\text{student}][\text{slot}]$ é igual ao valor absoluto do número de aulas do aluno *student* naquele *slot* (N) menos um (preenchimento da matriz O do solver).
- Os alunos com unidades curriculares de apenas um ano não podem ter sobreposições de aulas.
- No pior dos casos, existe uma sobreposição de duas aulas num mesmo slot.

Para atender aos desejos dos orientadores, foi criada uma nova *feature* de restrições ao solver já existente:

- É possível alterar dinamicamente a capacidade das salas para uma determinada UC, permitindo maior flexibilidade na geração dos horários.

Capítulo 11

Conclusão

Concluído o projeto, pensamos que este foi uma mais valia para a aprendizagem de novos conceitos que não são abordados no curso. Por outro lado, foi possível obter mais conhecimento acerca do estudado anteriormente na licenciatura, nomeadamente solvers.

Esperamos que terminado o mesmo, este possa hoje estar mais completo e otimizado para a utilização pela direção de Curso da Licenciatura em Ciências da Computação na distribuição dos alunos pelas diferentes UC's.