

Faculdade de Engenharia da Universidade do Porto



Projeto 1 AED 23/24

Sistema de gerência de horários da LEIC

Algoritmos e Estruturas de Dados FEUP-L.EIC011-2023/2024-1S:

Regente: Pedro Nuno Ferreira da Rosa da Cruz Diniz

Regente: Ana Paula Cunha da Rocha

Grupo G33 - Estudantes & Autores:

David dos Santos Carvalho Ranito up202206312@fe.up.pt

Pedro Daniel Freitas João up202204962@fe.up.pt

Tiago Freitas Torres up202208938@fe.up.pt

Data de entrega:

03/11/2023

Índice

Introdução.....	3
Estruturas de dados utilizadas	4
Estruturação de classes	5
Informação relativa ao curso	5
Pedidos	6
Inicialização do programa	6
Interface	6
Funcionalidades	7
Consultas	7
Opções de ordenação	10
Opções de filtragem	10
Pedidos	11
Reversão de pedidos	12
Término do programa	12
Conclusão.....	12

Introdução

A elaboração deste projeto surge no âmbito da unidade curricular Algoritmos e Estruturas de Dados, desenvolvida pelos alunos da turma 3, grupo G33, do 2º ano da Licenciatura em Engenharia Informática e Computação pela Faculdade de Engenharia da Universidade do Porto.

O objetivo deste projeto é o desenvolvimento de um sistema para gerir horários já elaborados, e incluir várias funcionalidades relacionadas com os horários (visualização, modificação, etc.). Para isso, primeiramente é necessário visualizar e realizar um esquema de todo o sistema, tal como as relações entre cada objeto, e averiguar as estruturas de dados mais adequadas, de modo a potencializar as funcionalidades a desenvolver.

No final do projeto, o objetivo é apresentar um sistema de gerência de horários correto, eficiente e de fácil utilização.

Estruturas de dados utilizadas

Na maior parte do programa foram usados *sets* e *vectors*. Para armazenar os dados foram utilizados *sets*, pois o *set* não permite elementos duplicados e a complexidade temporal de inserção num *set* é $O(\log(n))$ enquanto se fossem utilizados *vectors* teria de ser visto se já existe o elemento antes de o adicionar, que para um *vector* a complexidade temporal é $O(n)$.

Para a interface foram usados maioritariamente *vectors*, porque é necessário ordenar os elementos por uma ordem específica (de acordo com as opções de ordenação escolhidas) e a função *sort* da STL não pode ser utilizada com *sets*.

Para o histórico de pedidos aceites foi usado uma *stack* devido à sua política LIFO que ajuda a manter uma ordem cronológica.

Ao reverter o último pedido aceite, o pedido é retirado do topo e desfeito. Sempre que o programa termina, o *stack* é guardado num ficheiro e, quando volta a ser executado, os pedidos são lidos do ficheiro para uma *queue*. O programa refaz os pedidos da *queue*, de forma a voltar ao estado em que estava na última execução, sem recorrer à alteração dos ficheiros originais.

Estruturação de classes

Informação relativa ao curso

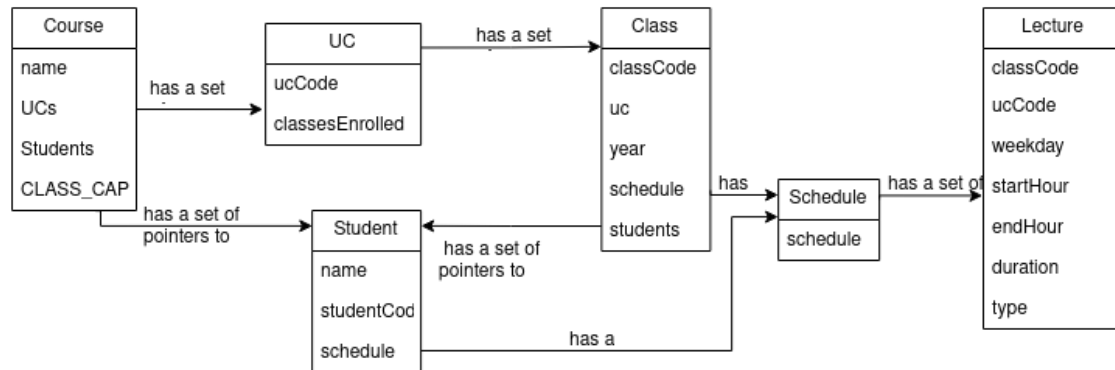


Figura 1- Diagrama das classes referentes à informação relativa ao curso

Course é a classe global que representa um curso e contém toda a informação sobre o mesmo.

Foi colocado um set de Students no Course para obter com mais rápido acesso os estudantes de todo o curso sem ter de procurar por UCs ou Classe. Os pointers são utilizados para que seja possível ter o mesmo Student em várias Classes. Ao realizar uma modificação, esta aplica-se ao mesmo objeto, isto é, o mesmo estudante, que se encontra em todas as turmas em que está inscrito.

A estruturação de classes implementada tem como objetivo ser de fácil entendimento e também eficiente implementação.

Pedidos

Para os pedidos foi criada uma classe Request e classes hereditárias representativas de cada tipo de pedido. Desta forma foi possível armazenar todos os pedidos na mesma estrutura de dados.

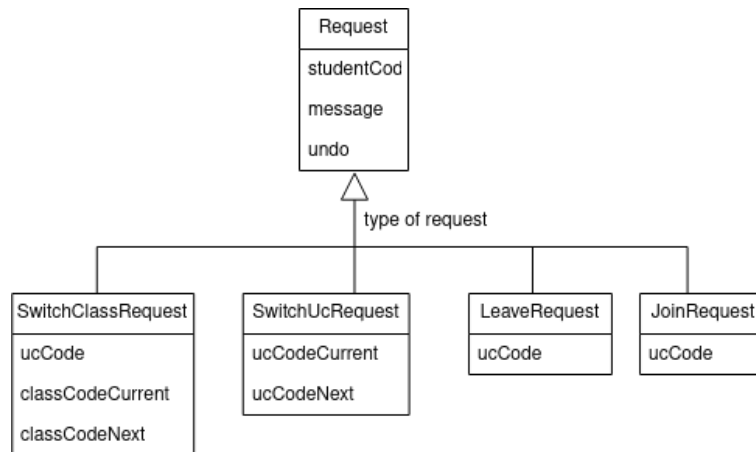


Figura 2 – Diagrama das classes referentes aos pedidos

Inicialização do programa

Ao inicializar o programa são extraídos dos ficheiros de texto todos os dados relativos ao curso e armazenados num objeto da classe Course, também são inseridos numa queue os pedidos realizados no passado para que possam ser executados novamente.

A leitura dos ficheiros de texto é realizada pela classe Auxiliar.

Interface

O projeto contém uma interface de interação entre o utilizador e o programa através da classe Terminal, que é responsável por gerir a execução do programa conforme os inputs do utilizador.

Todas as funcionalidades são apresentadas ao utilizador através de menus na consola (Ubuntu). Todas as opções possíveis em cada menu têm uma string ou caractere associados, que devem ser introduzidos pelo utilizador para as escolher, através da consola.

```
*****SCHEDULE LEIC*****

CONSULTS
0 - Consult the UC's of a year
1 - Consult the classes of a UC
2 - Consult the classes of a student
Consult the students of:
3 - a course
4 - a year
5 - a UC
6 - a class
Consult the schedule of:
7 - a student
8 - a class
9 - a UC
10 - Consult the number of students in at least n UC's
11 - Consult course occupation

REQUESTS
12 - Request to join a UC
13 - Request to leave a UC
14 - Request a UC switch
15 - Request a class switch

Press 'q' to quit.
Press the number corresponding the action you want.
_
```

Figura 3 - Menu inicial do programa

Funcionalidades

As funcionalidades estão divididas em dois principais grupos, as consultas e os pedidos.

Os dados necessários para as consultas e pedidos são introduzidos pelo utilizador, porém devem ser dados corretos, uma vez que o programa tem um comportamento inesperado quando introduzidos erradamente (ex. trocar estudante de uma turma em que este não está inscrito).

Consultas

Consultar as unidades curriculares de um ano

Input: o ano a ser consultado.

```
These are the UC's in year 1
L.EIC001 L.EIC002 L.EIC003 L.EIC004 L.EIC005 UP001

Press 'm' to go back to the main menu.
Press 'q' to quit.
_
```

Figura 4 - Output da consulta das unidades curriculares de um ano

Consultar as turmas de uma unidade curricular

Input: o código da unidade curricular a ser consultada.

```
These are the classes of UC L.EIC011
2LEIC01 2LEIC02 2LEIC03 2LEIC04 2LEIC05 2LEIC06
2LEIC07 2LEIC08 2LEIC09 2LEIC10 2LEIC11 2LEIC12
2LEIC13 2LEIC14 2LEIC15
Press 'm' to go back to the main menu.
Press 'q' to quit.
```

Figura 5 - Output da consulta das turmas de uma unidade curricular

Consultar as turmas de um estudante

Input: o código do estudante a ser consultado.

```
These are the classes of student 202020897
      UC      |      Class
-----|-----
L.EIC001      |      1LEIC01
L.EIC002      |      1LEIC01
L.EIC013      |      2LEIC13
L.EIC021      |      3LEIC11
L.EIC023      |      3LEIC11
L.EIC024      |      3LEIC11
L.EIC025      |      3LEIC11
```

Figura 6 - Output da consulta das turmas de um estudante

Consultar os estudantes do curso, de um ano, de uma unidade curricular ou de uma turma

Esta consulta tem opções de ordenação da tabela.

Input: para a consulta de estudantes de um ano, o ano a ser consultado, para a consulta de estudantes de uma unidade curricular, o código da unidade curricular e para a consulta de estudantes de uma turma, o código da unidade curricular e o código da turma.

```
These are the students registered in class 1LEIC01 in UC L.EIC001
  Code  | Name
-----|-----
202020897 | Francisca
202020642 | Sofia
Press 's' for sorting options.
Press 'm' to go back to the main menu.
Press 'q' to quit.
```

Figura 7 - Output da consulta dos estudantes de uma turma

Consultar o horário de um estudante, de uma turma e de uma unidade curricular

Input: Para a consulta do horário de um estudante, o código do estudante, para a consulta do horário de uma turma, o código da unidade curricular e o código da turma e para a consulta do horário de uma unidade curricular, o código da unidade curricular.

This is the schedule of UC L.EIC011						
HOUR	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
08:00 - 09:30		L.EIC011 (T)				
09:30 - 10:00		2LEIC09				
09:00 - 09:30			L.EIC011 (TP)			
09:30 - 10:00			2LEIC12			
10:00 - 10:30						
10:30 - 11:00		L.EIC011 (T)				
11:00 - 11:30		2LEIC01				
11:30 - 12:00			L.EIC011 (TP)			
12:00 - 12:30			2LEIC04			
12:30 - 13:00						
13:00 - 13:30						
13:30 - 14:00						
14:00 - 14:30						
14:30 - 15:00				L.EIC011 (T)	L.EIC011 (TP)	
15:00 - 15:30	L.EIC011 (TP)			2LEIC09	2LEIC13	
15:30 - 16:00	2LEIC11					
16:00 - 16:30				L.EIC011 (TP)		
16:30 - 17:00				2LEIC01	L.EIC011 (TP)	
17:00 - 17:30					2LEIC05	
17:30 - 18:00	L.EIC011 (TP)					
18:00 - 18:30	2LEIC08			L.EIC011 (TP)		
18:30 - 19:00				2LEIC09		
19:00 - 19:30						
19:30 - 20:00						
20:00 - 20:30						
20:30 - 21:00						
21:00 - 21:30						
21:30 - 22:00						

OVERLAPPED LECTURES			
UC	WEEKDAY	HOUR	CLASS
L.EIC011 (T)	Tuesday	08:00 - 09:00	2LEIC10
L.EIC011 (T)	Tuesday	08:00 - 09:00	2LEIC11
L.EIC011 (T)	Tuesday	08:00 - 09:00	2LEIC12
L.EIC011 (T)	Tuesday	08:00 - 09:00	2LEIC13
L.EIC011 (T)	Tuesday	08:00 - 09:00	2LEIC14
L.EIC011 (T)	Tuesday	08:00 - 09:00	2LEIC15
L.EIC011 (TP)	Tuesday	08:30 - 10:30	2LEIC02
L.EIC011 (T)	Tuesday	10:30 - 11:30	2LEIC02
L.EIC011 (T)	Tuesday	10:30 - 11:30	2LEIC03
L.EIC011 (T)	Tuesday	10:30 - 11:30	2LEIC04
L.EIC011 (T)	Tuesday	10:30 - 11:30	2LEIC05
L.EIC011 (T)	Tuesday	10:30 - 11:30	2LEIC06
L.EIC011 (T)	Tuesday	10:30 - 11:30	2LEIC07
L.EIC011 (T)	Tuesday	10:30 - 11:30	2LEIC08
L.EIC011 (TP)	Tuesday	10:30 - 12:30	2LEIC15
L.EIC011 (T)	Wednesday	10:00 - 11:00	2LEIC01
L.EIC011 (T)	Wednesday	10:00 - 11:00	2LEIC02
L.EIC011 (T)	Wednesday	10:00 - 11:00	2LEIC03
L.EIC011 (T)	Wednesday	10:00 - 11:00	2LEIC04
L.EIC011 (T)	Wednesday	10:00 - 11:00	2LEIC05
L.EIC011 (T)	Wednesday	10:00 - 11:00	2LEIC06
L.EIC011 (T)	Wednesday	10:00 - 11:00	2LEIC07
L.EIC011 (T)	Wednesday	10:00 - 11:00	2LEIC08
L.EIC011 (T)	Thursday	14:30 - 15:30	2LEIC10
L.EIC011 (T)	Thursday	14:30 - 15:30	2LEIC11
L.EIC011 (T)	Thursday	14:30 - 15:30	2LEIC12
L.EIC011 (T)	Thursday	14:30 - 15:30	2LEIC13
L.EIC011 (T)	Thursday	14:30 - 15:30	2LEIC14
L.EIC011 (T)	Thursday	14:30 - 15:30	2LEIC15
L.EIC011 (TP)	Thursday	15:30 - 17:30	2LEIC03
L.EIC011 (TP)	Thursday	15:30 - 17:30	2LEIC06
L.EIC011 (TP)	Thursday	17:30 - 19:30	2LEIC10
L.EIC011 (TP)	Friday	14:00 - 16:00	2LEIC14
L.EIC011 (TP)	Friday	16:00 - 18:00	2LEIC07

Figura 8 - Output da consulta do horário de uma unidade curricular

Consulta dos estudantes inscritos em pelo menos n unidade curriculares

Esta consulta tem opções de ordenação da tabela.

Input: número mínimo de unidade curriculares

These are the students registered in registered in at least 7 UC's	
Code	Name
202027102	Amelia
202043252	Bento
202022002	Bianca
202024807	Bruna
202025487	Eunice
202025402	Filipa
202020897	Francisca
202062037	Joao Rodrigo
202054642	Jose Nelson
202020217	Leonor
202053452	Manuel Carlos
202023532	Marta
202024552	Nicole
202040362	Raul
202024212	Valentina

Press 's' for sorting options.
Press 'm' to go back to the main menu.
Press 'q' to quit.

Figura 9 - Output da consulta de estudantes inscritos em pelo menos n unidades curriculares

Consulta da ocupação do curso

Esta consulta tem opções de ordenação e de filtragem da tabela

This course (LEIC) has 711 students registered.

Year	UC	Class	Occupation
2	L.EIC011	2LEIC01	19
2	L.EIC011	2LEIC02	22
2	L.EIC011	2LEIC03	22
2	L.EIC011	2LEIC04	21
2	L.EIC011	2LEIC05	21
2	L.EIC011	2LEIC06	22
2	L.EIC011	2LEIC07	21
2	L.EIC011	2LEIC08	22
2	L.EIC011	2LEIC09	22
2	L.EIC011	2LEIC10	23
2	L.EIC011	2LEIC11	23
2	L.EIC011	2LEIC12	21
2	L.EIC011	2LEIC13	24
2	L.EIC011	2LEIC14	23
2	L.EIC011	2LEIC15	23

Press 's' for sorting options.
Press 'f' for filtering options.
Press 'm' to go back to the main menu.
Press 'q' to quit.

Figura 10 - Output da consulta da ocupação das turmas de uma unidade curricular

Opções de ordenação

A ordenação da tabela pode ser feita através de qualquer coluna e de forma ascendente ou descendente.

```
SORTING OPTIONS
Type the number of the column (starting at 0) to order by: 0
Type 0 for descending order or 1 for ascending order: _
```

Figura 11 - Output das opções de ordenação

Opções de filtragem

As opções de filtragem da tabela permitem escolher quais as colunas mostradas e permitem especificar um elemento para uma coluna de modo a ser mostrada informação só para esse elemento.

```
FILTERING OPTIONS
Columns to display (0 - don't display, 1 - display):
  Year: 1
  UC: 1
  Class: 1
Filtering options ('all' to show all elements of the column):
  Year: all
  UC code: all
  Class code: all_
```

Figura 12 - Output das opções de filtragem

Pedidos

Pedido para um estudante se inscrever numa unidade curricular

Este pedido só é aceite se o estudante estiver inscrito em 6 ou menos unidades curriculares e se a unidade curricular tiver vagas e se for o caso, a turma escolhida é a que tiver menos estudantes de forma a equilibrar o número de alunos inscritos em cada turma dentro dessa unidade curricular e que simultaneamente não cause um conflito no horário do estudante (aulas TP ou PL sobrepostas).

Pedido para um estudante sair de uma unidade curricular

Este pedido é sempre aceite.

Pedido para um estudante fazer uma troca de unidades curriculares

Este pedido é aceite se se verificarem as condições do Pedido para um estudante se inscrever numa unidade curricular com a exceção da condição de estar inscrito em 6 ou menos

Pedido para um estudante fazer uma troca de turmas para a mesma unidade curricular

Este pedido é aceite se a turma em que o estudante quer ser inscrito tiver vaga, se não perturbar o equilíbrio da ocupação das turmas da unidade curricular em causa (diferença entre o número de estudantes por turma tem de ser 4 ou menos) e se não causar um conflito no horário do estudante.

```
Enter the student code: 202020897
Enter the UC code: L.EIC001
Enter the code of the class you are currently registered: 1LEIC01
Enter the code of the class you want to switch to: 1LEIC02_
```

Figura 13 - Output da formulação de um pedido de troca de turmas

```
Your request was accepted.
Student 202020897 switched from class 1LEIC01 to class 1LEIC07 for the UC L.EIC001
Press 'm' to go back to the main menu.
Press 'q' to quit.
```

Figura 14 - Output da decisão após avaliação do pedido

Reversão de pedidos

O último pedido aceite pode ser revertido.

```
The most recent request is: Student 202020897 switched from class 1LEIC01 to class 1LEIC07 for the UC L.EIC001.  
Are you sure you want to undo this request? (y/n)_
```

Figura 15 - Output da reversão do último pedido

Término do programa

O programa pode ser terminado pelo utilizador a qualquer momento. Assim que o utilizador escolher terminar o programa a classe Auxiliar fica encarregue de guardar num ficheiro de texto os pedidos realizados.

Conclusão

De modo geral, os objetivos iniciais foram cumpridos, foram aplicados e aprimorados os conhecimentos obtidos na frequência desta disciplina. Também foi adquirida experiência no desenvolvimento de programas de maior complexidade e extensão.

Este projeto permitiu o uso de várias estruturas de dados lecionadas neste semestre, assim como a exploração das relações entre elas e desenvolver a capacidade de resolução de problemas uma vez que estes foram surgindo e foi necessário alterar certos elementos para os resolver, por exemplo, mudar alguns *sets* de classes para *sets* de *pointers* da respetiva classe.

Concluindo, todos os objetivos determinados foram cumpridos e foi obtido um programa funcional com bastante versatilidade.