

PROJETO AED: GESTOR DE HORÁRIOS



ANTÓNIO ROMÃO, 202108704

FRANCISCO CAMPOS, 202108735

TOMÁS GASPAR, 202108828

DESCRIÇÃO DO PROBLEMA



- Pretende-se criar um sistema de Gestor de Horários da LEIC, que realize principalmente as 3 seguintes tarefas:
 1. Ler dados fornecidos e criar classes para os organizar logicamente, e armazenar;
 2. Criar funções de listagem (parcial e total) e ordenação, que permitam filtrar e organizar os dados obtidos;
 3. Criar, avaliar, e responder de forma adequada aos pedidos efetuados para alteração das turmas do utilizador.
- Para além disso, criar uma interface gráfica confortável à utilização do programa.

DESCRIÇÃO DA SOLUÇÃO



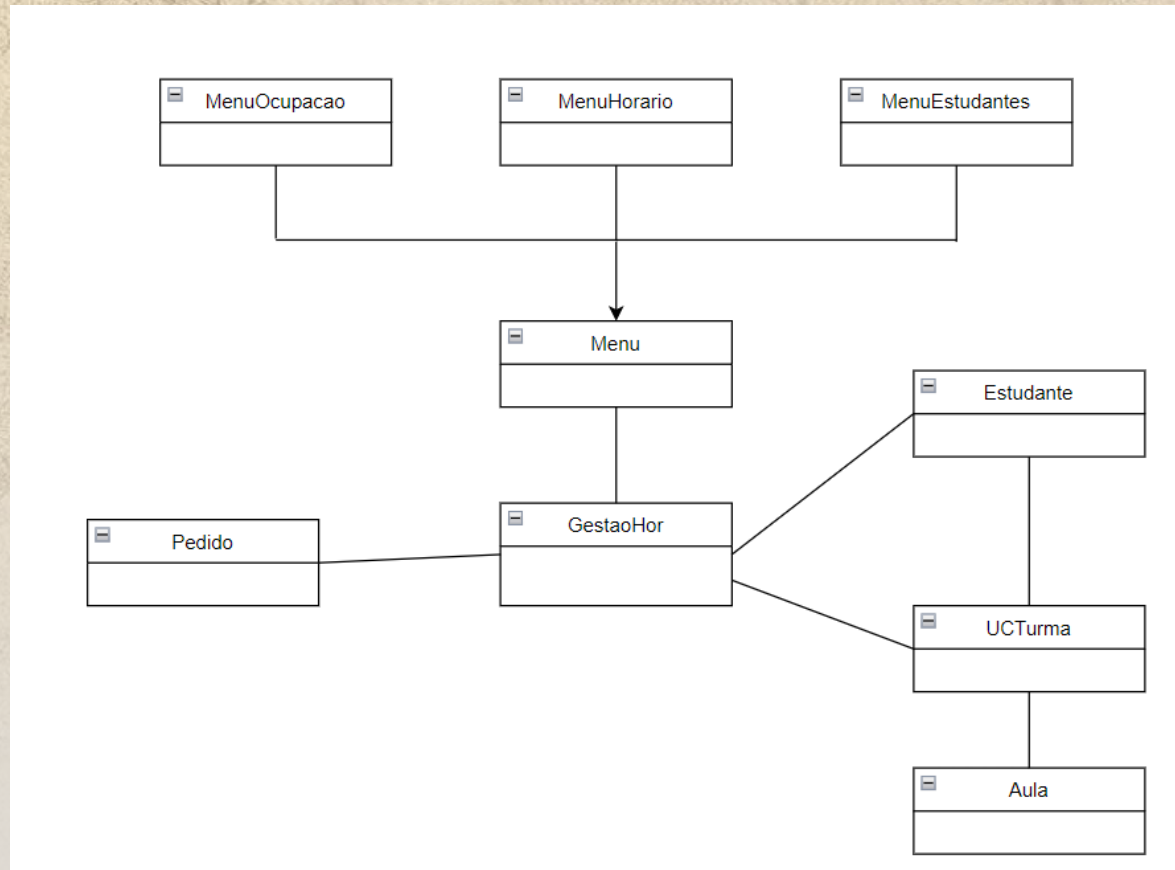
- Respondendo aos passos anteriores:
 1. Criar uma classe GestaoHor que gere os vários estudantes presentes no ficheiro students_classes.csv, e atribuir a cada Turma/UC as respetivas aulas (ficheiro classes.csv).
 2. Criar uma classe Menu, onde o utilizador pode interagir com o Gestor de Horários, e, após a recolha de dados em I., executar as funções listagem e ordenação que desejar.
 3. Na classe GestaoHor, criar uma fila de Pedidos para serem tratados de acordo com a ordem de entrada no sistema.



ALGORITMOS RELEVANTES

1. Para a recolha de dados, tiramos proveito da organização dos ficheiros `classes.csv` e `students_classes.csv` para otimizar o processo;
2. Criar um sistema de comunicação entre funções do Menu, para trocar o output produzido no ecrã;
3. Para cuidar dos casos em que o utilizador pede para mudar trocar de turma/UC, ou alterar várias turma/UC, decidimos criar partições nos pedidos (ou seja, a nossa fila passou a armazenar vetores de Pedidos, em vez de pedidos por si só). Uma adição ou remoção simples dava origem a um vetor de tamanho 1, uma troca de turma/UC 2, e de vários turma/UC n;

DIAGRAMA DE CLASSES



FUNCIONALIDADES IMPLEMENTADAS



GestaoHor	
Tratamento de Dados	Tratamento de Pedidos
getDataStudent	ocupacaoTurmasUC
getDataAula	addPedidos
	processPedidos
	classConflict
	removeStudentUCClass
	addStudentUCClass

FUNCIONALIDADES IMPLEMENTADAS



Menu:

`start()`

`alterarMenu();`

FUNCIONALIDADES IMPLEMENTADAS



MenuHorario:

bool horarioMenu();

- void searchHorarioByNum;
- void searchHorarioByName;
- void searchHorarioTurma;
- void printHorario

MenuOcupacao:

ocupacaoMenu();

- ocupacaoTurmasMenu
 - ocupacaoDeterminadaTurmaUC
 - printOcupacaoTurmasUC
- ocupacaoAnosMenu
 - ocupacaoDeterminadoAno
 - ocupacaoAnos
- ocupacaoUCMenu
 - ocupacaoDeterminadaUC
 - printOcupacaoUCs
 - ocupacaoUCs
 - ocupacaoUCsAno

MenuEstudantes:

bool estudantesMenu();

- void estudantesTurmaUC
- void estudantesAno
- void estudantesUC
- void estudantesMaisDeNUCs

MELHOR FUNCIONALIDADE



Processo de Pedidos (processPedido):

- Bom tratamento de erros/conflitos ao adicionar a uma UC/Turma
- Implementação extra: Estudante não pode estar em 2 turmas da mesma UC
- Error Log File: Guarda tentativas-erro de alteração das UC/Turma de um Estudante num ficheiro com uma descrição percetível
- Coerência com as restantes funções do programa
- Pedidos tratados em fila (que só é processada quando se sai do menuAlterar)

PROBLEMAS E PRESTAÇÃO DO GRUPO



Principais Problemas:

- Problema Organização Inicial
- Merge conflicts (Github)
- Utilizar BST antes de ser lecionada
- Organização de código

Prestação do Grupo:

- Francisco Campos e Tomás Gaspar estiveram mais responsáveis pela concretização do Código
- António Romão esteve responsável pela Documentação do Projeto

EXEMPLOS DE EXECUÇÃO



```
type 'q' to quit
=====
1
-> ESTUDANTES

    1 - Estudantes em determinada turma a determinada UC
    2 - Estudantes em determinado ano
    3 - Estudantes em determinada UC
    4 - Estudantes com mais de n UCs

type 'q' to quit, 'r' to return
=====
2
Ano: 3
Ordenação: 1) N° Estudante, 2) Alfabética
1
201951157 - Manuel Agostinho
201960762 - Gustavo Joao
201964162 - Jose Leopoldo
201978187 - Jose Ivo
202020897 - Francisca
202021662 - Luana
202023362 - Carminho
202024212 - Valentina
202024977 - Angelina
202025827 - Caetana
202026252 - Dulce
202027102 - Amelia
202027552 - ...
```

```
-> HORÁRIO

    1 - Estudante (pesquisa por número)
    2 - Estudante (pesquisa por nome)
    3 - Turma

type 'q' to quit, 'r' to return
=====
1
Número de Estudante: 202028717
UC: L.EIC003 | Turma: 1LEIC04
Thursday 8 2 TP
Friday 10 1 T
Wednesday 9.5 1 T
UC: L.EIC011 | Turma: 2LEIC11
Monday 14.5 2 TP
Thursday 14.5 1 T
Tuesday 8 1 T
UC: L.EIC012 | Turma: 2LEIC11
Tuesday 10.5 2 TP
Wednesday 10.5 1 T
Friday 18 1 T
```


EXEMPLOS DE EXECUÇÃO



- https://drive.google.com/file/d/1w3KDeil8CKSE25oOxp8rZsW773-12KFJ/view?usp=share_link

OBSERVAÇÕES:



Quando acaba o PPT, mas não tem um
meme

