

Trabalho 1: Algoritmo e estruturas de dados

Realizado por:

João Fernandes - up202108044

Luis Duarte - up202108734

Andreia Silva - up202108769

Descrição do problema

Criação de um sistema que permita:

- Gerir e manipular os horários da LEIC;
- Processar os pedidos de alteração de turma;
- Alteração de turmas de um estudante;
- Visualização do horário de um estudante/turma;
- Verificar os estudantes presentes numa turma, etc;

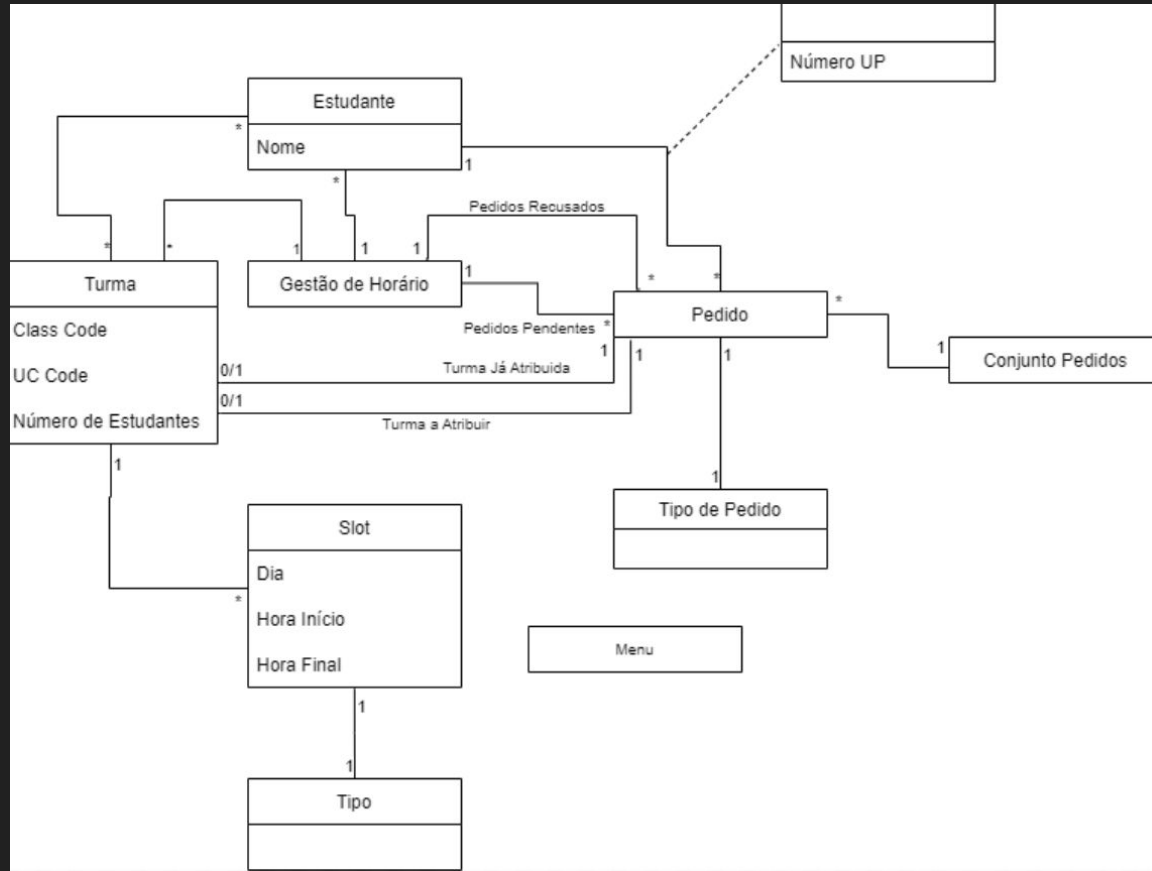
Descrição da solução (parte 1)

- Iniciamos com a leitura de ficheiros, colocando os estudantes num set e as turmas numa lista.
- Cada estudante terá como atributos a lista de turmas em que está inscrito e as turmas terão a lista de aulas associadas.
- Os pedidos são efetuados através da linha de comandos e adicionados a uma lista.
- Relativamente ao menu, são apresentadas as opções de pesquisa e, conforme a preferência do utilizador em termos de critérios de pesquisa e de ordenação, são apresentadas como listas, etc...
- Caso haja algum input em falso, o programa informa que é inválido e corre o menu outra vez.

Descrição da solução (parte 2)

- Ao processá-los serão analisados, verificando conflitos ou desequilíbrios de turmas.
- Na eventualidade de serem recusados, são adicionados à lista de pedidos recusados.
- Na função de verificação de conflito é criado um novo horário e posteriormente confirma-se se este é possível.
- Na função de desequilíbrio o número de alunos de cada turma é conferido, testando se o pedido em questão não fará haver diferenças de 4 ou mais alunos.

UML das classes usadas



Funcionalidades implementadas

- Ler os ficheiros para memória e atualizá-los ao fechar o programa;
- Ver horário de estudante ou turma, ou até todas as turmas de uma UC;
- Ver vagas disponíveis em cada turma de uma UC para não causar desequilíbrio;
- Ver nº de estudantes numa turma e quais estudantes eles são;
- Ver estudantes com mais de n UCs por quantidade ou por ordem alfabética;
- Fazer pedidos de alteração de adicionar, remover e alterar. Estes pedidos são verificados por conflitos nos horários e por desequilíbrios na turma.

Destaque de uma funcionalidade

- A leitura dos ficheiros é cross-platform (funciona entre unix-based e windows), usando principalmente macros

```
.....  
//o windows usa \\ para os diretórios enquanto o unix usa /  
#ifndef _WIN32  
    std::ifstream ucs = std::ifstream("../recursos/classes_per_uc.csv");  
    std::ifstream estudantes_turma = std::ifstream("../recursos/students_classes.csv");  
    std::ifstream turma_horario = std::ifstream("../recursos/classes.csv");  
#else  
    std::ifstream ucs = std::ifstream("../\\recursos\\classes_per_uc.csv");  
    std::ifstream estudantes_turma = std::ifstream("../\\recursos\\students_classes.csv");  
    std::ifstream turma_horario = std::ifstream("../\\recursos\\classes.csv");  
#endif
```

Destaque de uma funcionalidade

- Entre o Windows e o Linux a maneira que fazem uma nova linha num ficheiro de texto é diferente: (windows usa `\r\n` e linux usa só `\n`)

```
std::string& trim_string(std::string& str) {  
    #ifndef _WIN32  
        /*  
         * O windows usa \r\n para sinalizar o fim de uma linha, Enquanto que o linux só usa \n.  
         * Ou seja ao ler ficheiros com o formato do windows no linux vamos ter um caracter a mais  
         * e vai induzir em erros na comparação de turmas.  
         */  
        if(str.find_first_of('\r') != std::string::npos)  
            str = str.substr(0, str.size()-1);  
    #endif  
    return str;  
}
```


Destaque de outra funcionalidade

- O uso de smart-pointers ao guardar as turmas em memória, neste caso, o `std::shared_ptr`.
- Simplifica muitos pedidos de alteração de turma, só tendo de alterar uma vez os dados e garante melhor *memory-safety*.

Dificuldades no trabalho

- Fazer o algoritmo dos pedidos de alteração de turmas foi a parte mais difícil porque:
 - Um pedido de alteração pode causar desequilíbrio, mas outros pedidos mais à frente da lista podem resolver esse desequilíbrio (especialmente na troca de turmas).

Dificuldades no trabalho

- Uma má *git flow* levou a algumas complicações no trabalho:

