

GestorHorarios

Afonso Sousa 202007585

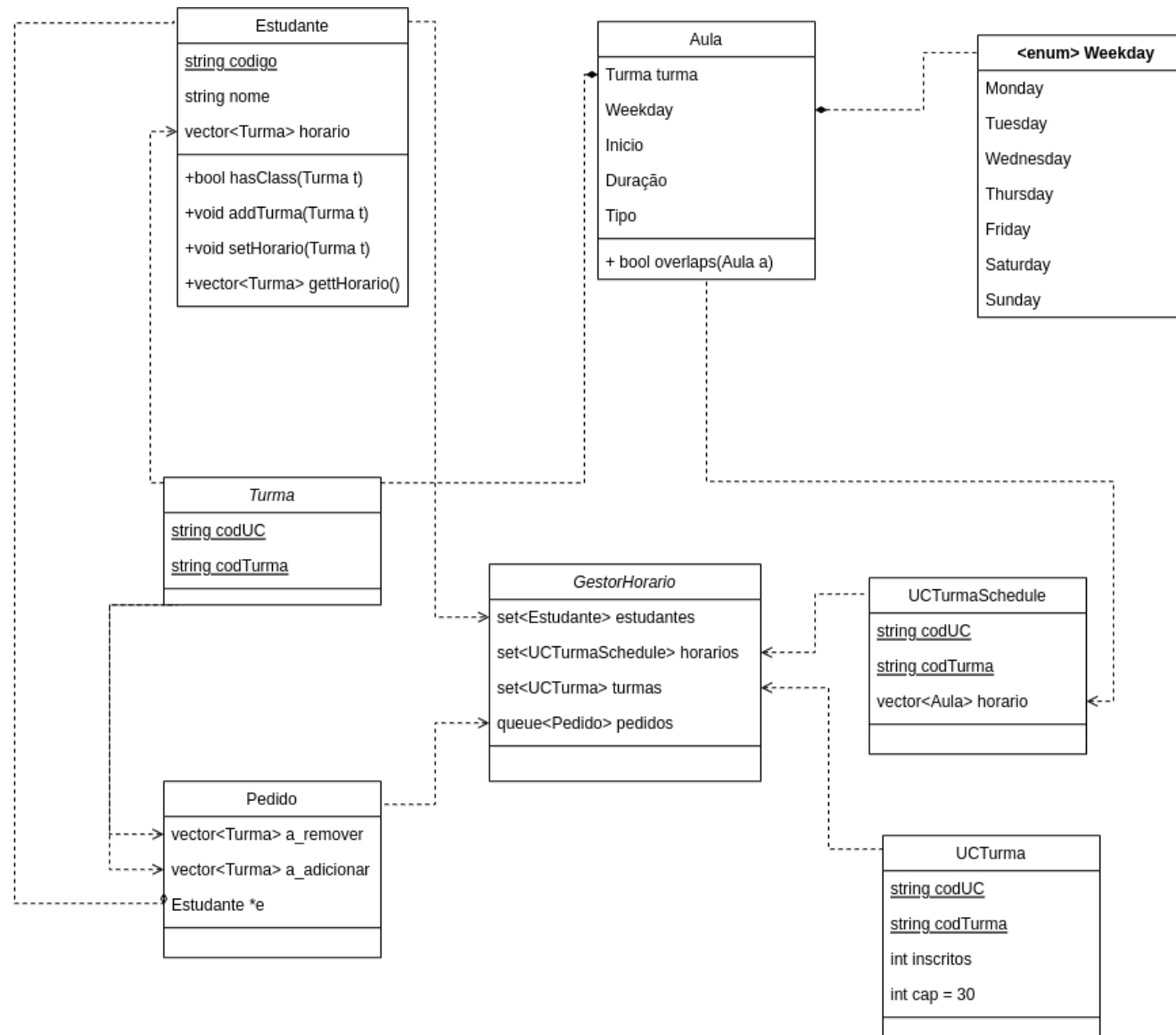
Problema: Gestor de Horários

- Desenvolver um sistema capaz de gerir horários previamente atribuídos. Este sistema deve ser capaz de :
 - Alterar o horário de um estudante
 - Consultar os dados de Estudantes e UCs

Solução

- Cada Estudante tem um horário composto pelas turmas em que está inscrito.
- Cada Turma tem um horário que contém as Aulas para essa turma.
- O utilizador pode criar um Pedido com as turmas a remover ou adicionar ao horário de um Estudante.
- Os Pedidos são processados, no fim, por ordem de criação e podem ser rejeitados caso a inscrição desequilibre as Turmas.

Diagrama de Classes



Listagem de Turmas

TURMAS

Código UC [0]	Código Turma [1]	N Inscritos [2]
L.EIC013	2LEIC01	22
L.EIC013	2LEIC02	26
L.EIC013	2LEIC03	26
L.EIC013	2LEIC04	26
L.EIC013	2LEIC05	23
L.EIC013	2LEIC06	25
L.EIC013	2LEIC07	24
L.EIC013	2LEIC08	23
L.EIC013	2LEIC09	26
L.EIC013	2LEIC10	23
L.EIC013	2LEIC11	26
L.EIC013	2LEIC12	26
L.EIC013	2LEIC13	25
L.EIC013	2LEIC14	26
L.EIC013	2LEIC15	26
L.EIC013	2LEIC16	26

COMMANDS:

```
sort NUM          Ordenar por propriedade NUM [0-2] (1 == 0)
sortd NUM         Ordenar por propriedade (descendente) NUM [0-2] (1 == 0)
search NUM "STR"  Restringir list a elementos que contenham STR na propriedade NUM [0-2]
horario            Selecionar turma e imprimir horario
reset              Limpar critérios
back
```

\$ _

Listagem de Estudantes

ESTUDANTES

Código[0]	Nome [1]	N Inscrições [2]
202034072	Afonso	5
202050562	Manuel Afonso	5
202050647	Jose Afonso	1

COMMANDS:

sort <u>NUM</u>	Ordenar por propriedade <u>NUM</u> [0-2]
sortd <u>NUM</u>	Ordenar por propriedade (descendente) <u>NUM</u> [0-2]
search <u>NUM</u> " <u>STR</u> "	Restringir list a elementos que contenham <u>STR</u> na propriedade <u>NUM</u> [0-2]
searh_turma	Selecionar turma e filtrar
horario <u>NUM</u>	Imprimir horario do estudante <u>NUM</u>
select <u>NUM</u>	Selecionar estudante e criar pedido <u>NUM</u> [Nº Estudante]
reset	Limpar critérios
back	

\$

Criação de Pedido

TURMAS

Código UC [0]	Código Turma [1]
L.EIC025	3LEIC12

(202080397):
(a)dicionar | (r)emover | (b)ack(| (s)ave >

Menu de criação de pedido após seleção do estudante

GestorHorario	
Fazer Pedido	[1]
Listagens	[2]
Save (processar pedidos)	[3]
Exit	[4]

(#):

Menu Principal, onde podem ser efetivados os pedidos

Ordenação das Listas para apresentação

- É feita com recurso ao mergesort: complexidade **$O(n \log n)$**

```
get_comp(i, cmp);  
utils::mergesort(v, cmp);
```

```
void GestorHorarios::get_comp(int i, std::function<bool(const Estudante &, const Estudante &)> &cmp) {  
    switch(i){  
        case 0: cmp = [ ](const Estudante &p1, const Estudante &p2){ return p1.get_codigo() > p2.get_codigo(); }; break;  
        case 1: cmp = [ ](const Estudante &p1, const Estudante &p2){ return p1.get_nome() > p2.get_nome(); }; break;  
        case 2: cmp = [ ](const Estudante &p1, const Estudante &p2){ return (p1.getHorario().size() > p2.getHorario().size()); }; break;  
        case 3: cmp = [ ](const Estudante &p1, const Estudante &p2){ return p1.get_codigo() < p2.get_codigo(); }; break;  
        case 4: cmp = [ ](const Estudante &p1, const Estudante &p2){ return p1.get_nome() < p2.get_nome(); }; break;  
        case 5: cmp = [ ](const Estudante &p1, const Estudante &p2){ return (p1.getHorario().size() < p2.getHorario().size()); }; break;  
        default: throw std::invalid_argument("NUM outside range");  
    }  
}
```


Dificuldades encontradas

- Modelar uma solução adequada e implementá-la linearmente