

GESTÃO DE HORÁRIOS

Adriano Machado /Francisco Pires da Ana /José Pedro Evans

DESCRIÇÃO DO PROBLEMA

Este projeto tem como objetivo o desenvolvimento de um sistema capaz de ajudar na gestão de horários (alteração, visualização). Para este efeito, deve-se escolher as estruturas de dados mais apropriadas e eficientes.

Tipos de dados utilizados

● BST

Para o armazenamento de pedidos de trocas de turmas usamos uma Queue.

Método	Complexidade Temporal
Insert()	$O(\log N)$
Erase(valor)	$O(\log N)$
Size()	$O(1)$

● Queue

No armazenamento dos estudantes usamos uma BST.

Método	Complexidade Temporal
Push()	$O(\log N)$
Empty()	$O(1)$
Pop()	$O(\log N)$
Front()	$O(1)$

Tipos de dados utilizados

- **Vector**

Esta estrutura de dados foi usada, por exemplo, no armazenamento de horários ou de aulas de um determinado estudante.

Implementamos um método, `binarySearchSchedules` que aplica pesquisa binária no vetor de horários e retorna o índice no horário da UcClass pretendida. Este método apresenta complexidade temporal $O(\log N)$.

Método	Complexidade temporal
<code>Push_back()</code>	$O(1)$
<code>At()</code>	$O(1)$
<code>Size()</code>	$O(1)$

Classes

Student

id: str
name: str
classes: vector<UcClass>

UcClass

ucId: str
classId: str

Slot

weekDay: str
startTime: float
endTime: float
type: str

Request

student: Student
desiredClass: UcClass

ClassSchedule

ucClass: UcClass
slots: vector<Slot>
students: set<Student>

ScheduleManager

students: set<Student>
schedules: vector<ClassSchedule>
requests: queue <Request>
rejectedRequests: vector<Request>

Funcionalidades implementadas

Leitura dos dados fornecidos

O método readFiles() é chamado. Este por sua vez chama os métodos:

- createSchedules() – Lê o ficheiro classes_per_uc.csv e adiciona ao vetor de horários(schedules) objetos da classe ClassSchedule com a devida UcClass, mas com o vetor de slots e sets de estudantes vazios;
- setSchedules() – Lê o ficheiro classes.csv e adiciona aos objetos ClassSchedule do vetor schedules o devido slot;
- createStudents() – Lê o ficheiro students_classes.csv e adiciona objetos Student ao set Students; adiciona também a cada objeto ClassSchedule do vetor schedules os estudantes daquela turma;

1	UcCode,ClassCode
2	L.EIC001,1LEIC01
3	L.EIC001,1LEIC02

Fig 1. – classes_per_uc.csv

1	ClassCode,UcCode,Weekday,StartHour,Duration,Type
2	1LEIC01,L.EIC001,Monday,10.5,1.5,TP
3	1LEIC02,L.EIC001,Thursday,9.5,1.5,TP

Fig 2. – classes.csv

1	StudentCode,StudentName,UcCode,ClassCode
2	201920727,Ines,L.EIC001,1LEIC05
3	201920727,Ines,L.EIC002,1LEIC05

Fig 3. – students_classes.csv

Funcionalidades implementadas

Visualização dos dados fornecidos

A partir do menu principal temos a capacidade de:

- Ver o horário de um determinado estudante;
- Ver o horário de uma turma;
- Ver as aulas de uma determinada cadeira;
- Ver os alunos inscritos numa turma a uma dada cadeira (ordenados alfabeticamente ou pelo número mecanográfico);
- Ver os alunos inscritos numa cadeira;
- Submeter um pedido de alteração de turma;
- Processar os pedidos;

Funcionalidades implementadas

Pedidos de alteração de horários

Destaque de Funcionalidade

Dificuldades encontradas / Esforço