

Projeto 1

Horários L.EIC

A elaboração de horários para as aulas do L.EIC é uma tarefa complexa. O objetivo deste projeto não é a criação dos horários, mas sim o desenvolvimento de um sistema para gerir os horários depois de elaborados. O sistema deve incluir várias funcionalidades relacionadas com os horários, tais como alteração, pesquisa, visualização, ordenação, listagem, entre outras.

O ficheiro **schedule.zip** contém informações reais sobre os horários do L.EIC para o 1º semestre do ano letivo de 2022/2023 com dados anónimos dos alunos. Segue-se uma descrição detalhada do conjunto de dados fornecido. Note-se que o horário de um aluno pode sobrepor-se a aulas se este não for TP nem PL. Ou seja, pode haver sobreposição de aulas entre T e TP, entre T e T, e entre T e PL. Um exemplo específico de um horário é apresentado abaixo.

2022 - A 1S 1T 2T SP

Horas	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
08:00 - 08:30					<u>MD (T)</u> <u>1LEIC01</u>	
08:30 - 09:00						
09:00 - 09:30	<u>AM I (T)</u> <u>1LEIC01</u>	<u>AM I (TP)</u> <u>1LEIC01</u>			<u>B001</u>	
09:30 - 10:00	<u>B001</u>	<u>B328</u>	<u>FP (T)</u> <u>1LEIC01</u>	<u>FP (TP)</u> <u>1LEIC01</u>		
10:00 - 10:30			<u>B003</u>	<u>B301</u>	<u>FP (T)</u> <u>1LEIC01</u>	
10:30 - 11:00	<u>ALGA (TP)</u> <u>1LEIC01</u>	<u>ALGA (T)</u> <u>1LEIC01</u>	<u>FSC (TP)</u> <u>1LEIC01</u>		<u>B001</u>	
11:00 - 11:30				<u>FSC (T)</u> <u>1LEIC01</u>	<u>MD (TP)</u> <u>1LEIC01</u>	
11:30 - 12:00	<u>B338</u>	<u>B001</u>	<u>B335</u>			
12:00 - 12:30				<u>B002</u>	<u>B327</u>	
12:30 - 13:00						
13:00 - 13:30						
13:30 - 14:00						
14:00 - 14:30			<u>PUP (TP)</u> <u>1LEIC01</u>			
14:30 - 15:00			<u>B102</u>			
15:00 - 15:30						
15:30 - 16:00						
16:00 - 16:30						
16:30 - 17:00						
17:00 - 17:30						
17:30 - 18:00						
18:00 - 18:30						
18:30 - 19:00						
19:00 - 19:30						
19:30 - 20:00						

Conjunto de dados

O conjunto de dados fornecido está disponível em **schedule.zip** sob a forma de três ficheiros csv (*valores separados por vírgulas*), conforme descrito abaixo.

- **Ficheiro turmas_por_uc.csv:** contém as turmas existentes em cada unidade curricular (UC).

A primeira linha inclui os cabeçalhos: *UcCode* (código da unidade curricular), *ClassCode* (código da turma).

```
UcCode,ClassCode
L.EIC001,1LEIC01
L.EIC001,1LEIC02
```

- **Ficheiro classes.csv:** contém os horários das aulas.

A primeira linha inclui os cabeçalhos: *ClassCode* (código da aula), *UcCode* (código da unidade curricular), *Weekday* (dia da semana), *StartHour* (hora de início da aula), *Duration* (duração da aula em horas), *Type* (tipo de aula: T, TP, PL).

```
ClassCode,UcCode,Weekday,StartHour,Duration,Type
1LEIC01,L.EIC001,Monday,10.5,1.5,TP
1LEIC02,L.EIC001,Quinta-feira,9.5,1.5,TP
1LEIC03,L.EIC001,Terça-feira,9,1.5,TP
```

- **Ficheiro alunos_turmas.csv:** contém as turmas dos alunos de cada UC.

A primeira linha inclui os cabeçalhos: *StudentCode* (código do aluno), *StudentName* (nome do aluno), *UcCode* (código da unidade curricular), *ClassCode* (código da turma).

```
StudentCode,StudentName,UcCode,ClassCode
202025232,Iara,L.EIC002,1LEIC05
202031607,Gisela,L.EIC004,1LEIC08
202031607,Gisela,L.EIC005,1LEIC08
202079037,Jose Jesualdo,L.EIC023,3LEIC08
```

Conceptualmente, uma *turma* é identificada pelo código da UC, dia da semana, hora de início, duração e tipo (T/TP/PL). Um *horário* é composto por várias turmas. Um *aluno é identificado* por um nome, e tem um horário associado. Uma *turma é identificada* por um código e tem um determinado horário.

Declaração de Trabalho (SoW)

Neste projeto, é convidado a utilizar as estruturas de dados vetor, lista, pilha, fila e árvore de pesquisa binária (pode utilizar outras estruturas, se necessário). Estas estruturas servirão de base para as tarefas seguintes e devem ser utilizadas no seguinte contexto:

1. Ler e analisar os dados fornecidos. Isto inclui o carregamento dos dados analisados nas estruturas de dados mais apropriadas para as funcionalidades necessárias. Preste muita atenção às estruturas de dados lineares e hierárquicas que selecionar para cada cenário, de modo a poder tirar partido das vantagens e desvantagens de cada estrutura de dados de forma adequada.
2. Desenvolver um sistema de gestão do calendário. O sistema deve incorporar um conjunto completo de funcionalidades de gestão do calendário, incluindo as especificadas para este projeto e quaisquer outras consideradas relevantes. Deve ter um menu de fácil utilização que permita ao utilizador definir os critérios desejados. O menu deve apresentar as funcionalidades disponíveis e os respectivos outputs de forma clara, organizada e lógica, facilitando a utilização do programa e a interpretação direta dos resultados.
3. Efetuar várias listagens dos dados, de forma total ou parcial. Ter em consideração que o utilizador pode querer obter informação filtrada do sistema. Neste sentido, é também importante que a informação listada seja ordenada de forma lógica, de modo a facilitar ainda mais a utilização do programa. Como fonte de inspiração, pode considerar as seguintes listagens:
 - i. Consultar o horário de um determinado aluno ou turma;
 - ii. Consultar os alunos de uma determinada turma, curso ou ano;
 - iii. Consultar o número de estudantes inscritos em pelo menos n UCs;
 - iv. Consultar a ocupação classe/ano/UC (ordenação por UC, ordem crescente ou decrescente, ...);
 - v. Consultar as UCs com maior número de estudantes;
 - vi. ...
4. Processar pedidos de novas inscrições ou actualizações de inscrições existentes. Considere que os alunos podem precisar de se registar para uma mudança de UC ou de turma. Estes pedidos devem ser analisados antes de serem aceites e, por isso, devem ser armazenados em estruturas de dados adequadas para processar um pedido de cada vez ou todos de uma só vez.

Os pedidos podem ser de três tipos diferentes: *adicionar*, *remover* ou *trocar*. Um pedido de *adição/remoção* ocorre quando um aluno pretende entrar/sair de uma UC ou de uma turma. Por fim, um pedido de *troca* permite ao aluno trocar uma UC ou uma turma em que estava anteriormente inscrito por uma nova UC ou turma.

 - i. Para alterações de UC, considere as seguintes regras:
 1. Um estudante não pode estar inscrito em mais de 7 UCs em simultâneo;
 2. Deve haver pelo menos uma turma com vaga na nova UC;
 3. O horário resultante não entrará em conflito com o horário original do aluno.
 - ii. Para as mudanças de classe, considere as seguintes regras:
 - 1 - Um aluno não pode estar em mais do que uma turma em simultâneo numa dada UC;
 - 2 - Uma turma só pode aceitar um novo aluno se a sua capacidade não tiver sido excedida. Considere que existe um **limite máximo de capacidade** para as turmas;

3. uma turma só pode aceitar um novo aluno se o equilíbrio entre a ocupação das turmas não for perturbado. O equilíbrio da ocupação das turmas mantém-se quando a diferença entre o número de alunos de qualquer turma é **inferior ou igual a 4**;
4. Não há conflito entre o horário do aluno e o horário da nova turma.
5. Manter um registo cronológico de todas as alterações efectuadas no sistema. O sistema deve guardar os pedidos aceites pela ordem em que foram processados, permitindo aos utilizadores e administradores desfazer facilmente as acções mais recentes. Ter em consideração que a anulação de alguns pedidos pode violar as regras acima mencionadas, impossibilitando a operação de anulação. Nestes casos, o pedido de anulação deve ser recusado e o seu registo retirado do sistema.
6. Inclua a documentação das funções mais relevantes que implementou, gerada utilizando o Doxygen. Indique a complexidade temporal das funções ou algoritmos mais relevantes do seu programa.

Para além das funcionalidades acima mencionadas, o utilizador pode também implementar outras funcionalidades se as considerar relevantes.

Resultados esperados

O programa que desenvolver deve permitir o registo e a gestão de várias entidades diferentes e das suas relações, utilizando estruturas de dados lineares (vetor, lista, pilha, fila) e hierárquicas (árvore de pesquisa binária).

Neste sentido, é necessário prestar muita atenção aos seguintes pontos:

- Modelar as entidades relevantes num sistema **baseado em classes** com interacções adequadas entre as classes definidas;
- Utilizar de forma adequada as **estruturas de dados lineares** (vetor, lista, pilha, fila) e **hierárquicas**;
- Guardar informações importantes em **ficheiros** para utilização futura;
- Inclua **a documentação** do código implementado (gerado com Doxygen) e indique a **complexidade temporal** das funções ou algoritmos mais relevantes do seu programa.

O seu programa deve também permitir várias listagens diferentes com as seguintes condições:

- As listagens devem ser **totais e parciais** com critérios definidos pelo utilizador;
- Devem ser fornecidas ao utilizador diferentes opções **de ordenação**;
- As funcionalidades de listagem devem utilizar algoritmos **de pesquisa e de ordenação**.

Demonstração e apresentação

A preparação de uma apresentação concisa e objetiva é uma competência muito importante. Por conseguinte, deve estruturar uma breve apresentação de 15 minutos do seu trabalho, centrando-se nos seguintes pontos:

- Apresente o seu sistema de gestão de horários, utilizando exemplos ilustrativos para demonstrar as funcionalidades implementadas;
- Destacar e justificar as estruturas de dados utilizadas para representar o conjunto de dados fornecido, bem como as estruturas de dados utilizadas para implementar as funcionalidades solicitadas (modificação, pesquisa, visualização, ordenação, listagem, ...);
- Destaque os aspectos mais importantes da sua implementação.

Deve também elaborar uma **apresentação em PowerPoint** para apoiar a sua demonstração.

Instruções e prazo de entrega

Envie um ficheiro zip com o nome **AED2324_PRJ1_G<TN>.zip** no Moodle, onde **TN** representa os números da turma e do grupo (por exemplo, **G10** representa o grupo **0** da turma **2LEIC01**), com o seguinte conteúdo:

- Pasta de código, que contém o código fonte do programa.
- Pasta de documentação, contendo documentação HTML, gerada com Doxygen.
- Ficheiro de apresentação (**formato PDF**) que servirá de base à demonstração.

Não serão aceites envios 24 horas após o termo do prazo. As exceções aplicam-se a problemas técnicos justificados e documentados.

O prazo termina a 3 de novembro de 2023 à meia-noite.