

### 1. Objetivo

Pretende-se desenvolver um programa que auxilie na gestão das aulas de um curso transmitidas *online*. Considera-se que o curso terá no máximo 100 estudantes inscritos.

### 2. Descrição do problema

O programa deverá registar a informação referente às unidades curriculares (UCs) do curso, e de todas as aulas lecionadas *online*. Estas aulas poderão também ser gravadas, devendo essa indicação ser dada no início da aula. A gravação de uma aula apenas será disponibilizada depois da sua realização.

Cada UC terá um regime de funcionamento (diurno ou pós-laboral), e poderá ter vários tipos de aulas (T, TP ou PL) lecionadas *online*. Considera-se que, numa UC, todas as aulas do mesmo tipo têm a mesma duração, e que não poderão decorrer em simultâneo várias aulas da mesma UC. As aulas do regime diurno funcionam no período [8h-18h], e as do regime pós-laboral funcionam no período [18h-24h].

As aulas *online* só poderão realizar-se caso sejam previamente agendadas. O agendamento das aulas *online*, bem como o registo do seu início e do seu final, será efetuado através de um menu de opções.

O (registo do) acesso de um estudante a uma aula também é efetuado através do menu. Se a aula estiver a decorrer, considera-se que o estudante assiste *online* à aula. Caso a aula já se tenha realizado, se existir a gravação, considera-se que o estudante acede à aula gravada (acesso *offline* à aula).

### 3. Descrição dos elementos

**UC:** Cada UC (unidade curricular) é caracterizada pelo código (único), pela designação, pelo tipo de UC (obrigatória ou opcional), pelo semestre (1 a 6) e pelo regime (diurno, pós-laboral) de funcionamento, pela quantidade prevista de cada tipo de aulas (T, TP e PL) *online*, e pela respetiva duração (em minutos). O curso terá, no máximo, 40 UCs.

**Aula *online*:** Cada aula *online* é caracterizada pela designação (única), pelo código da UC, pelo tipo de aula (T, TP ou PL), pelo nome do docente, pela data, pela hora de início, pela hora de fim, pelo estado da aula (agendada, a decorrer, realizada) e pela opção de gravação (sim, não).

## 4. Funcionalidades da aplicação a desenvolver

A aplicação, a desenvolver na linguagem C, deverá apresentar ao utilizador um menu de opções que disponibilize as seguintes funcionalidades:

- Inserir, alterar e listar a informação referente às UCs do curso. Deverão também ser listadas, para cada UC, a quantidade de aulas *online* (de cada tipo) realizadas, e a informação (tipo, data, hora de início) das aulas agendadas.
- Agendar aulas *online* e listar os dados de todas as aulas *online*. Deverão também ser listadas, para cada aula *online* realizada, a quantidade de estudantes presentes e a quantidade de acessos às gravações.
- Alterar aula agendada (eliminando ou alterando o agendamento).
- Registrar o início de uma aula *online* (previamente agendada), e o final de uma aula *online* (a decorrer).
- Registrar o acesso (*online* ou *offline*) de um estudante a uma aula (*online* ou gravada). A informação do acesso (número do estudante, identificação da aula e tipo de acesso) deverá ser acrescentado num log, em ficheiro de texto e em ficheiro binário.
- Armazenar (e ler) em ficheiro(s) binário(s) a informação referente às UCs e às aulas *online*.
- Apresentar no menu principal a seguinte informação: quantidade de UCs, quantidade de aulas agendadas, quantidade de aulas realizadas e quantidade de aulas gravadas.
- Apresentar a informação de uma aula *online* (selecionada pelo utilizador), incluindo a seguinte informação da respetiva UC: designação e quantidade de aulas por agendar.
- Apresentar (por ordem decrescente) o ranking das UCs considerando a quantidade total de acessos às aulas gravadas.
- Apresentar os seguintes dados estatísticos: quantidade média de presenças em cada aula *online* realizada, percentagem de UCs com aulas gravadas, UC(s) com a menor quantidade de aulas *online* realizadas, o tipo(s) de aula(s) com a maior quantidade de acessos às gravações, e aula(s) *online* realizada(s) há mais tempo.
- Apresentar uma listagem com a identificação (números) dos estudantes presentes numa aula *online* (indicada pelo utilizador).

## 5. Notas Importantes

- Para assegurar uma boa implementação do Projeto deverá previamente identificar as principais estruturas de dados a utilizar e esboçar o algoritmo.
- O Projeto deverá estar dividido em várias funções e dividido por vários ficheiros, de forma a facilitar a sua correta estruturação, implementação e entendimento, e para evitar a repetição de código.
- Deverão ser efetuadas todas as validações e proteções necessárias ao correto funcionamento do programa.
- O programa deverá apresentar mensagens de aviso/erro sempre que necessário.
- Deverá ser dada prioridade à implementação das funcionalidades requeridas, em detrimento da implementação de funcionalidades adicionais.
- A funcionalidade do programa é mais relevante do que a apresentação da informação.

## 6. Regras e informações complementares

1. A **Fase 1** do Projeto deverá ser realizado em **grupos de dois estudantes** do mesmo turno PL.
2. Prazo para inscrição do grupo (no moodle): **19 dezembro 2020**.
3. Prazos para **submissão do projeto** (através do moodle): **25 janeiro 2021**.
4. **O projeto só será considerado definitivamente submetido após obterem essa informação do moodle.**
5. Na **Fase 1** deverão ser entregues os seguintes elementos em formato digital:
  - O **projeto** (ficheiros de código fonte), que terá de **compilar** no gcc utilizando o IDE *Codeblocks*.
  - **Relatório** contendo obrigatoriamente a descrição e representação gráfica das estruturas de dados utilizadas, e a identificação das funcionalidades não desenvolvidas ou parcialmente/incorrectamente desenvolvidas.  
Poderá também conter a justificação de algumas opções de implementação tomadas.
6. A **Fase 2** do projeto é individual, realizando-se a **27 janeiro 2021**, e consiste na elaboração de código adicional à versão entregue na Fase 1.
7. Data prevista para afixação da nota do projeto: **18 fevereiro 2021** (nota final).
8. Poderão ser agendadas **defesas de projetos** quando surgirem dúvidas quanto à sua autenticidade.

## 7. Cotações

**Nota do Projeto (0-20) = 50% \* notaFase1 (mín de 8.0) + 50% notaFase2 (mín de 8.0)**

### **Critérios de correção da Fase 1 do Projeto e respetivas cotações (\*)**

a) Relatório contendo todos os elementos obrigatórios .....	2,5%
b) Estrutura de Menus (incluindo informação solicitada).....	5,0%
c) Gestão de UCs, e respetivas listagens .....	10,0%
d) Agendamento e listagem de aulas .....	10,0%
e) Alteração de aula agendada .....	7,5%
f) Registo de início/final de uma aula .....	5,0%
g) Registo do acesso de um estudante a uma aula (e armazenamento em <i>logs</i> ) .....	7,5%
h) Ficheiros (leitura e escrita de dados dos estudantes, das UCs e das aulas).....	7,5%
i) Informação de aula <i>online</i> .....	5,0%
j) Ranking de UCs .....	7,5%
k) Dados estatísticos (cálculo e apresentação).....	17,5%
l) Listagem de estudantes presentes numa aula.....	10,0%
m) Estrutura geral do programa, incluindo a adequada utilização de variáveis locais, constantes, funções, comentários, indentação do código, separação do código por ficheiros.....	5,0%

**(\*)** Com exceção da alínea a), **as cotações apenas serão aplicadas em projetos que compilem sem erros**, não se considerado o código comentado.

Apenas serão cotados os **relatórios contendo todos os elementos obrigatórios** (descrição e representação gráfica das estruturas de dados, e a identificação das funcionalidades não desenvolvidas ou parcialmente/incorrectamente desenvolvidas).

**Será anulado qualquer projeto total ou parcialmente plagiado**, podendo os estudantes envolvidos ser sujeitos a **procedimento disciplinar**.

É considerado plágio a cópia de textos, documentos, imagens ou código de outras fontes, e considerá-lo como próprio.

**Essa situação pode ocorrer quando é partilhado código entre grupos de projeto, ou quando o projeto é desenvolvido por elementos de vários grupos.**