

## 1. Objetivo

Pretende-se desenvolver um programa, em linguagem C, para que os estudantes possam treinar para as provas de escolha múltipla da unidade curricular de Programação I.

## 2. Descrição do problema

O programa deverá registar a informação referente às perguntas, aos estudantes e aos treinos realizados de preparação para as provas.

Cada treino deverá ser constituído por um conjunto de 4 a 10 perguntas pertencentes a uma determinada prova.

Durante o treino deverão ser apresentadas as perguntas, as respetivas 4 respostas possíveis, e registadas as opções efetuadas pelo estudante. No final do treino deverá ser registado o tempo utilizado, e apresentada a quantidade de respostas corretas, a identificação das respostas erradas, e a classificação obtida. Todas as perguntas têm a mesma cotação, e por cada resposta errada desconta-se 25% dessa cotação.

## 3. Descrição dos elementos

**Perguntas:** Cada pergunta é caracterizada por uma designação (única), pela questão, por 4 respostas possíveis (identificação e texto), pela identificação da resposta correta, pelo tipo de pergunta (estrutura de controlo, vetor ou ponteiro) e pela prova (1 a 4) a que pertence. Considera-se que, no máximo, existem 40 perguntas.

**Estudante:** Cada estudante é caracterizado por um número (único), nome, tipo (normal, trabalhador ou extraordinário), regime (diurno ou pós-laboral) e turno .  
Considera-se que, no máximo, existem 20 estudantes.

**Treinos:** Cada treino é caracterizado pelo número de registo (único e atribuído automaticamente), pelo número do estudante, pela data e hora de realização, pela duração (em minutos), pela prova (1 a 4) correspondente, e pelo estado (criado, concluído). Deverá também armazenar a seguinte informação: quantidade de perguntas, respostas corretas, respostas erradas e classificação.

**Respostas:** Cada resposta é caracterizada pela designação da pergunta, pelo número do treino e pela resposta selecionada pelo estudante. Em cada treino existem, no máximo, 10 respostas.

## 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, consultar, alterar e listar a informação referente às perguntas. Deverão também ser listados a quantidade de vezes que cada pergunta foi utilizada nos treinos e a percentagem de respostas corretas, e indicada a quantidade de perguntas existente para cada prova.
- Inserir, alterar, consultar e listar os dados dos estudantes. Deverão também ser listados a quantidade de treinos efetuados e a percentagem de respostas corretas.
- Consultar e listar os dados dos treinos.
- Registrar a criação de um treino indicando, entre outros, o número do estudante e a prova que pretende treinar. Poderá também selecionar o tipo de perguntas.
- Registrar a realização de um treino previamente criado, obtendo as respostas do estudante, apresentando os resultados.  
A informação referente ao treino deverá ser acrescentada num log, em ficheiro de texto, e num ficheiro binário.
- Eliminar os dados de um treino que não se encontre realizado.
- Apresentar o resultado obtido por um determinado estudante (indicado pelo utilizador) no último treino, e listar todos os treinos desse estudante.
- Apresentar (por ordem decrescente) o ranking dos estudantes de acordo com os resultados obtidos nos treinos.
- Apresentar no menu principal a seguinte informação: quantidade de estudantes, quantidade de perguntas, quantidade de treinos realizados, e quantidade de treinos a decorrer.
- Armazenar (e ler) em ficheiro(s) binário(s) a informação referente aos estudantes, às perguntas e aos treinos.
- Apresentar os seguintes dados estatísticos: a tempo médio de resposta a uma pergunta, quantidade de treinos realizados entre duas datas (indicadas pelo utilizador), percentagem de treinos efetuados por cada prova, pergunta com a maior quantidade de respostas erradas, e o tipo de perguntas com a menor percentagem de respostas corretas.
- Consultar as perguntas e as respostas registadas num determinado treino (indicado pelo utilizador), identificando as respostas corretas.

## 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.
- A implementação de funcionalidades adicionais poderá valorizar o projeto, no entanto deverá ser dada prioridade à implementação das funcionalidades requeridas.
- 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 indicação do grupo (no moodle): **10 dezembro 2019**.
3. Prazos para **submissão do projeto** (através do moodle): **13 janeiro 2020**.
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 4.9.2 utilizando o IDE *Codeblocks*.
  - **Relatório** contendo a descrição das estruturas de dados utilizadas e a sua representação gráfica, a justificação das opções de implementação tomadas e a identificação das funcionalidades não desenvolvidas.
6. A **Fase 2** do projeto é individual, realizando-se a **15 janeiro 2020**, e consiste na elaboração de código adicional à versão entregue na Fase 1.
7. Data prevista para afixação da nota do projeto: **4 fevereiro 2020** (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 a representação gráfica e a descrição das estruturas de dados utilizadas, a justificação das opções de implementação tomadas, e a identificação das funcionalidades não desenvolvidas.....5,0%
- b) Estrutura de Menus .....5,0%
- c) Gestão de perguntas, e respetivas listagens .....5,0%
- d) Gestão de estudantes, e respetivas listagens.....5,0%
- e) Consulta e listagem de treinos .....2,5%
- f) Registo da criação de um treino.....7,5%
- g) Registo de realização de um treino (e armazenamento em ficheiros) .....12,5%
- h) Eliminar informação de treino.....5,0%
- i) Listar informação sobre o último treino de um estudante .....7,5%
- j) Ranking de estudantes.....5,0%
- k) Ficheiros (leitura e escrita de dados dos estudantes, das perguntas e dos treinos) 7,5%
- l) Dados estatísticos (cálculo e apresentação).....17,5%
- m) Informação de treino .....10,0%
- n) Estrutura geral do programa, incluindo a adequada utilização de variáveis locais, constantes, funções, comentários para auxiliar na interpretação do código (incluindo funções), correta indentação do código .....5,0%

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

**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 por vezes ocorre quando estudantes de vários grupos se juntam para elaborar o projeto, ou quando partilham código.**