

## Enunciado

Recorrendo à linguagem de programação (ANSI) C, implemente uma aplicação que permita realizar a gestão da disciplina de Linguagens de Programação. Sempre que se justificar, a aplicação deve fazer uso de listas ligadas para armazenar conjuntos de dados em memória. Por outro lado, a aplicação não pode recorrer ao uso de variáveis globais.

A aplicação deve suportar três tipos de utilizadores: aluno, docente, e docente-regente. Os utilizadores devem autenticar-se perante a aplicação (número mecanográfico e palavra-passe) para poder tirar partido das respetivas funcionalidades. Para além dos dados de autenticação, os utilizadores devem ter associados um nome e tipo de utilizador.

De modo a simplificar a utilização da aplicação, esta deve, na sua inicialização, pré-preencher-se com o conjunto de docentes da disciplina (incluindo o docente-regente) e com um número mínimo de 10 alunos.

### Funcionalidades básicas

1. Funcionalidades do sistema:
  - (a) **Pré-preenchimento dos dados iniciais** (corpo docente e pelo menos 10 alunos).
2. Funcionalidades visíveis a todos os utilizadores:
  - (a) **Login.**
  - (b) **Logout (volta à vista inicial de login).**
  - (c) **Alterar dados próprios.** Esta funcionalidade permite apenas que um dado utilizador altere o seu nome e/ou a sua palavra-passe.
3. Funcionalidades visíveis apenas para utilizadores do tipo docente ou docente-regente:
  - (a) **Inserir utilizador.** Deve incluir os dados de autenticação.
  - (b) **Listar alunos por ordem de número mecanográfico.** Nesta funcionalidade, a palavra-passe dos utilizadores deve ser sempre omitida das listagens.
  - (c) **Remover utilizador.**
  - (d) **Alterar dados de utilizador.** Esta funcionalidade permite a alteração de todos os dados de um determinado utilizador.

## Funcionalidades intermédias

1. Funcionalidades visíveis apenas para utilizadores do tipo docente ou docente-regente:
  - (a) **Configurar listagem de alunos (ordenação).** Disponibilizar uma opção de configuração que permita escolher o modo de ordenação da listagem de alunos. Os modos de ordenação são: (i) através do número mecanográfico; e (ii) através do nome.
  - (b) **Listar alunos segundo opção de ordenação.** Esta funcionalidade depende da opção de ordenação da listagem de alunos (ver ponto anterior).
  - (c) **Configurar listagem de alunos (paginação).** Disponibilizar uma opção de configuração que permita escolher o modo de paginação da listagem de alunos. Os modos de paginação são: (i) sem paginação (todos os alunos); (ii) com paginação (neste modo, o utilizador tem de indicar o número de alunos a apresentar por página).
  - (d) **Listar alunos segundo opção de paginação.** Esta funcionalidade depende da opção de paginação da listagem de alunos (ver ponto anterior).

## Funcionalidades avançadas

Cada aluno terá de realizar duas provas de avaliação, teste teórico-prático (TTP) e um trabalho prático (TP). Para a realização destas provas, o aluno dispõe de duas modalidades: avaliação contínua ou exame final (época normal de exames). Em ambas as modalidades, no caso de não obter aprovação a uma ou mais provas, poderá realizar essas mesmas provas em exame de época de recurso. As regras para considerar um aluno aprovado são:

- Ter obtido em determinada época de avaliação um mínimo de 7,5 valores no TTP e de 9,5 valores no TP.
  - Ter uma nota final mínima de 9,5 valores, com a seguinte ponderação: 40% TTP + 60% TP.
1. Funcionalidades visíveis apenas para alunos:
    - (a) **Escolher modalidade de avaliação** (contínua ou exame final). Os alunos que não tenham escolhido uma modalidade de avaliação estão por defeito incluídos na modalidade de exame final. Caso um aluno tenha optado pela avaliação contínua, não pode posteriormente alterar esta decisão. Após o primeiro lançamento de notas, não é mais possível alterar a modalidade de avaliação.
    - (b) **Consultar situação atual** (reprovado, aprovado, ou avaliação em curso). Um aluno que, em qualquer época de avaliação, tenha apenas realizado uma das provas de avaliação tem uma situação de avaliação em curso. Esta funcionalidade deve sempre exibir as notas que um aluno já obteve até ao momento (em todas as provas de avaliação) e a respetiva nota final.
  2. Funcionalidades visíveis apenas para utilizadores do tipo docente-regente:
    - (a) **Lançar notas.** O lançamento de notas deve obedecer à ordem de realização das provas, i.e., 1.º TTP avaliação contínua, 2.º TP avaliação contínua, 3.º TTP exame época normal, etc. Em cada lançamento de notas, devem ser apresentados ao docente-regente apenas os alunos elegíveis para a referida prova. Na época de recurso, apenas os alunos que tenham reprovado na avaliação anterior são elegíveis para o lançamento de notas.
  3. Funcionalidades visíveis apenas para utilizadores do tipo docente ou docente-regente:
    - (a) **Listar classificações dos alunos.** Acrescentar à já existente funcionalidade de listagem de alunos a indicação das notas obtidas nos diferentes momentos de avaliação e da nota final (caso já seja possível ser calculada).
    - (b) **Estatísticas.** Apresentar, para cada época de avaliação para a qual já tenha havido lançamento das notas TTP e TP:
      - i. Número de aprovados.
      - ii. Percentagem de aprovados face aos alunos elegíveis.

## Observações

- Na resolução deste trabalho prático é proibido o uso de variáveis globais e obrigatório, sempre que se justificar, o uso de Listas Ligadas. Os trabalhos que não respeitem estas regras serão avaliados com zero valores.
- O presente enunciado não representa uma especificação fechada do trabalho passível de ser desenvolvido. Assim, os alunos são instigados a implementarem funcionalidades que, apesar de não especificadas, possam ser relevantes para o problema apresentado.
- Este trabalho pode ser realizado em grupo. Um grupo pode ter no máximo dois alunos que se encontrem inscritos a Linguagens de Programação.
- O trabalho terá de ser entregue até às 24:00 horas do dia 8 de Janeiro de 2012. Para tal, cada aluno terá de submeter o resultado do seu trabalho na página da unidade curricular Linguagens de Programação na plataforma Moodle. **Todos os alunos terão de submeter o trabalho, mesmo que o/a colega de grupo já o tenha feito. Os alunos que não respeitem esta regra serão avaliados com zero valores.** Os ficheiros que compõem o trabalho terão de possuir, na forma de comentário, a identificação (número mecanográfico e nome completo) de cada aluno que o realizou. O ficheiro submetido terá de possuir um nome na forma <número mecanográfico do aluno>.zip (exemplo: 8101234.zip).
- O trabalho pressupõe uma apresentação e respectiva defesa do mesmo perante um júri constituído por um ou mais docentes da unidade curricular de Linguagens de Programação, a decorrer durante as aulas marcadas entre os dias 11 e 14 de Janeiro de 2012. Cada aluno terá entre dez a quinze minutos para apresentar como achar mais conveniente o trabalho por si desenvolvido, ao que poderá seguir-se uma defesa do trabalho durante quinze a trinta minutos. A apresentação e/ou defesa poderão ser interrompidas a qualquer instante assim o júri o entenda. Durante a defesa do trabalho há a possibilidade do júri pedir ao aluno que realize pequenas alterações ao trabalho ou implementações de funcionalidades não suportadas pelo trabalho.

## CrITÉrios de avaliação

CrITÉrio	Valores (0-20)
Funcionalidades básicas	5
Funcionalidades intermédias	2
Funcionalidades avançadas e/ou funcionalidades não especificadas	4
Documentação e legibilidade do código	1
Separação de conceitos e modularidade do código	2
Usabilidade da aplicação (inclui aspetos relativos ao <i>deployment</i> da aplicação)	1
Apresentação do trabalho	2
Defesa do trabalho	3