

# Gestão de Exames do DEI

---

Projeto de POO

## Estrutura geral do programa

O programa possui 13 classes claramente diferenciadas. A classe Pessoa é abstrata e apresenta um método abstrato (menu) a ser executado por cada uma das suas classes. Nessas classes (Aluno, Docente e NaoDocente) o método é apresentado de maneira diferente com base no tipo de pessoa. Nessas classes o utilizador pode adicionar, remover ou alterar dados sobre cada uma das pessoas. O programa também apresenta as classes Curso, Disciplina, Exame e Sala em que são desenvolvidos métodos para a criação, edição e remoção destes objetos. Já a classe Estatísticas é responsável por apresentar as estatísticas do programa ao utilizador de acordo com as especificações do enunciado. A classe FicheiroTexto é responsável pela receção dos dados da classe Estatísticas e a escrita dos mesmos, caso assim o utilizador pretenda, num ficheiro de estatísticas, estatisticas.txt. Finalmente a classe FicheiroObjetos é responsável pela leitura e escrita dos ficheiros de objetos relativamente às pessoas, exames, cursos e salas. Com as classes vão também ser enviados ficheiros com dados para efeitos de teste.

## Descrição da estrutura de dados

Como foi dito anteriormente o programa possui treze classes cada uma com a sua estrutura de dados. Começando pela classe Docente ela possui uma String que representa a área de investigação do docente em causa. Dados como o nome, email, número e categoria são obtidos nas super class a que a classe Docente está relacionada. O mesmo processo ocorre para a classe NaoDocente, que possui apenas uma String para representar o cargo e para a classe Aluno, que possui três dados: um long para representar o seu número de aluno, um inteiro para representar o ano em que se encontra e uma string para representar o estatuto do aluno. A classe Curso possui duas strings, uma para o nome do curso e outra para o grau que o curso confere, um inteiro que representa a duração do curso e possui também uma ArrayList que identifica as disciplinas que o curso possui. É uma ArrayList de disciplinas. A classe Sala possui apenas uma string para identificar o nome da sala e um inteiro para identificar a sua ocupação máxima. A classe Disciplina possui uma string para identificar o nome da disciplina, um objeto da classe Docente para indentificar o docente responsável, uma ArrayList de objetos da classe Docente para identificar os docentes auxiliares e uma ArrayList de objetos da classe Aluno para identificar os alunos inscritos à disciplina. A classe Exame possui um objeto da classe Disciplina para representar a disciplina associada ao exame, um objeto da classe Date para representar a data do exame na forma (yyyy-MM-dd HH:mm), um inteiro para representar a duração do exame e um objeto da classe Sala para representar a sala onde o exame vai decorrer. A classe Departamento é responsável pela inicialização e conclusão do programa e por isso possui estruturas de dados, na forma de ArrayList, capazes de guardar todos os dados que o programa lê quando inicia ou fecha. Esta classe é responsável pelo "preenchimento" destas ArrayList e a sua utilização nos menus que cada classe possui. A classe FicheiroTexto possui um objeto da classe BufferedWriter para escrever no ficheiro e um objeto da classe FileWriter que é usado pelo objeto da classe BufferedWriter para conseguir escrever no ficheiro de texto. A classe

FicheiroObjetos possui um objeto da classe `ObjectInputStream` e outro objeto da classe `ObjectOutputStream` para executar, respetivamente, a leitura e escrita dos ficheiros.

## Ficheiros usados

Relativamente aos ficheiros são criados, se não existirem, quatro ficheiros de objetos para as classes Sala, Curso, Exame e Pessoa e um ficheiro de texto para a classe Estatisticas onde se guarda toda a informação em relação às estatísticas do programa, caso o utilizador pretenda que assim seja. Em conjunto com este relatório vão também ser enviados ficheiros de teste para as classes anteriormente mencionadas. Os ficheiros contêm os seguintes dados:

### Utilizadores:

Docentes (na forma nome; email; número de docente; categoria; área de investigação):

- a; [a@dei.uc.pt](mailto:a@dei.uc.pt) ; 1; assistente; redes
- b; [b@dei.uc.pt](mailto:b@dei.uc.pt); 2; auxiliar; iia
- c; [c@dei.uc.pt](mailto:c@dei.uc.pt); 3 ;auxiliar; sistemas de informação
- d; [d@dei.uc.pt](mailto:d@dei.uc.pt); 4; catedrático; comunicação e telemática
- e; [e@dei.uc.pt](mailto:e@dei.uc.pt); 5; associado; sistemas de informação
- f; [f@dei.uc.pt](mailto:f@dei.uc.pt); 6; assistente; redes
- g; [g@dei.uc.pt](mailto:g@dei.uc.pt); 7; associado; telemática
- h; [h@dei.uc.pt](mailto:h@dei.uc.pt); 8; auxiliar; sistemas operativos
- i; [i@dei.uc.pt](mailto:i@dei.uc.pt); 9; assistente; compressão de dados
- j; [j@dei.uc.pt](mailto:j@dei.uc.pt); 10; catedrático; redes

Não Docentes (na forma nome; email; número de funcionário; categoria; cargo):

- k; [k@dei.uc.pt](mailto:k@dei.uc.pt); 11; assistente operacional; secretaria
- l; [l@dei.uc.pt](mailto:l@dei.uc.pt); 12; assistente técnico; financeiro
- m; [m@dei.uc.pt](mailto:m@dei.uc.pt); 13; técnico de informática; apoio técnico
- n; [n@dei.uc.pt](mailto:n@dei.uc.pt); 14; técnico superior; secretaria
- o; [o@dei.uc.pt](mailto:o@dei.uc.pt); 15; especialista em informática; financeiro

Aluno (na forma nome; email; número de aluno; ano de inscrição; regime):

- p; [p@dei.uc.pt](mailto:p@dei.uc.pt); 16; 1; normal
- q; [q@dei.uc.pt](mailto:q@dei.uc.pt); 17; 3; normal
- r; [r@dei.uc.pt](mailto:r@dei.uc.pt); 18; 2; atleta
- s; [s@dei.uc.pt](mailto:s@dei.uc.pt); 19; 1; dirigente associativo
- t; [t@dei.uc.pt](mailto:t@dei.uc.pt); 20; 2; trabalho-estudante

Salas (na forma nome; ocupação):

Anfiteatro A; 250;

Anfiteatro B; 50

Anfiteatro C; 50

Curso (na forma nome; duração; grau; disciplinas):

Engenharia Informática; 3; Licenciatura; SO, IRC, TI, POO, SD

Disciplina (na forma nome; regente; docentes auxiliares; alunos):

SO; a; b,c,f,e; 16,17,18,19,20

IRC; h; i,j,c,f; 16,17,18,19,20

TI; d; g,h,e,a; 16,17,18,19,20

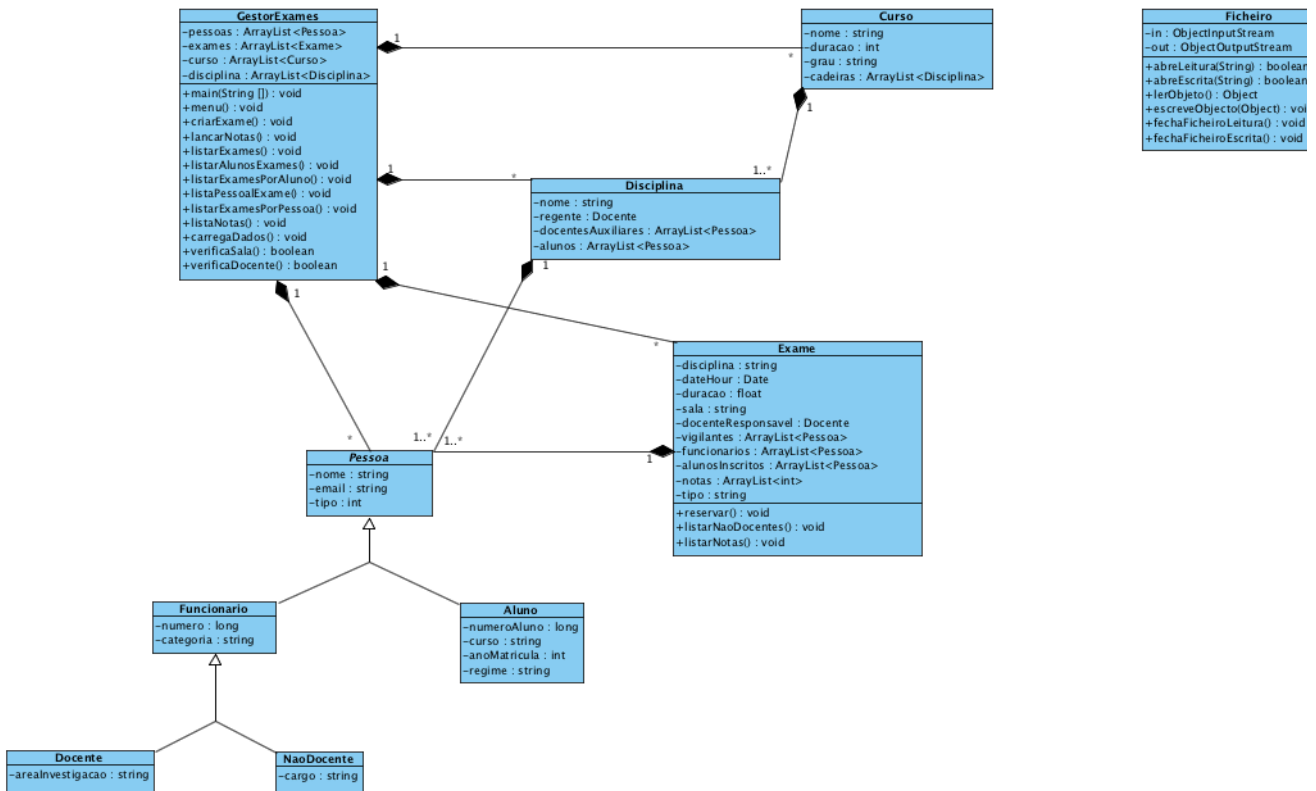
POO; c; f,a,b,i; 16,17,18,19,20

SD; e; b,f,h,j; 16,17,18,19,20

## Funcionamento do programa

Assim que o programa inicia os ficheiros que são anteriormente mencionados são lidos e os seus dados são guardados nas respetivas ArrayList. Caso os ficheiros não existam são criados na pasta do programa. Após a leitura dos ficheiros o programa apresenta ao utilizador um menu onde este pode escolher entre 5 opções: Exame, Pessoal, Cursos, Salas e Estatísticas. A opção Exame permite ao utilizador criar um exame, lançar as notas de um exame, alterar um parâmetro de um exame ou remover um exame. A opção Pessoal apresenta por sua vez mais um menu onde o utilizador pode escolher entre Docente, NaoDocente e Aluno. Para cada uma destas hipóteses o utilizador pode adicionar, alterar ou remover uma Pessoa. A opção Cursos permite ao utilizador adicionar, remover ou alterar um curso e respetivamente as suas disciplinas. A opção Salas permite ao utilizador adicionar, remover ou alterar uma sala. Finalmente, a opção Estatísticas permite ao utilizador obter dados sobre exames, cursos e pessoas. A cada opção o utilizador pode guardar os dados que lhe são apresentados no ficheiro de texto criado para o efeito. A qualquer momento do programa o utilizador pode voltar ao menu inicial, bastando apenas premir 0. Quando o utilizador quiser sair do programa todos os dados do programa são guardados nos respetivos ficheiros e o programa termina com sucesso.

## Diagrama inicial



## Diagrama final

