



## **Projecto**

### **Programação Orientada a Objectos, Estruturas de Dados, Manipulação de Ficheiros, Acesso a Bases de Dados Relacionais, Interfaces Gráficas**

## **1. Introdução**

Neste projecto, pretende-se a implementação de uma aplicação completa em linguagem Java, com componente gráfica para interacção com o utilizador, recorrendo ao paradigma de Programação Orientada a Objectos e com suporte para bases de dados relacionais.

Espera-se o desenvolvimento e aplicação de conhecimentos e competências relacionadas com as temáticas abordadas na disciplina, com especial incidência para a:

- **programação orientada a objectos**;
- **manipulação de bases de dados relacionais através de JDBC**;
- uso de **ficheiros** para armazenamento de dados de forma persistente;
- bem como o desenvolvimento de **interfaces para interacção com o utilizador** (em modo gráfico).

## **2. Objectivos**

Existe um conjunto de requisitos que o projecto deve respeitar e implementar:

### **2.1. Descrição global da aplicação**

- [ R1 ] Corrigir e completar a aplicação iniciada no projecto 1, cumprindo os respectivos requisitos.
- [ R2 ] Não é necessário apresentar mensagens de boas-vindas no arranque e de encerramento, nem estatísticas de utilização, como data, duração ou número de execuções da aplicação.

#### **Gestão de acesso e utilizadores**

- [ R3 ] Além dos atributos definidos no projecto 1, os **utilizadores** possuem uma foto, que pode ser inserida ou alterada em qualquer momento.

Caso o utilizador não defina a sua foto, deve existir uma imagem por defeito/genérica no sistema.

- [ R4 ] **(Para grupos)** No momento de registo deve ser enviado um email a informar que o registo foi realizado.

#### **Ações**

- [ R5 ] Em todos os processos deve existir uma caixa para inserir observações.
- [ R6 ] Após autenticação, deve surgir uma caixa de diálogo ou uma zona na janela principal em realce a indicar a existência de notificações, caso existam.

- [ R7 ] (**Para grupos**) Deve ser possível imprimir um extracto das acções de um processo (indicação das datas e dos utilizadores que realizaram acções num processo).

### **Listagens e pesquisas**

- [ R8 ] Todas as listagens devem incluir barras de deslocamento para permitir visualizar os resultados.

## **2.2. Gestão geral da aplicação**

### **Interacção com o utilizador**

- [ R9 ] Toda a interacção com o utilizador (apresentação de informação ou manipulação de dados) deverá ser realizada através de **interfaces gráficas** (componentes gráficas como janelas, painéis, caixas de diálogos, menus, abas, tabelas, botões, caixas de texto, etc.) disponíveis através da biblioteca gráfica JavaSwing / JavaFX.

A organização da interface e escolha de componentes fica ao critério do aluno, no entanto devem ser intuitivas, limpas e claras. Soluções eficientes de disposição e transição de componentes, respeitando as regras de interacção com o utilizador serão valorizadas.

- [ R10 ] Devem ser usadas componentes gráficas adequadas à informação a apresentar.

Por exemplo, na apresentação de dados estruturados (listagens) devem ser usadas listas (JList) ou tabelas (JTable) e não em texto simples (JLabel ou JTextField).

- [ R11 ] Os diversos formulários devem possuir uma componente de ajuda.

Sobre cada elemento da interface (e.g. campo de texto, botões) deve existir uma região onde sobrepondo o rato permita o aparecimento de uma nova janela com a informação de ajuda (*tooltips*).

### 3. Implementação

---

O programa deve ser implementado na linguagem Java. Lembre-se que é uma linguagem orientada a objectos, pelo que deverá ter em conta os seguintes aspectos:

- Cada classe deve gerir internamente os seus dados, pelo que deverá cuidar da protecção das suas variáveis e métodos.
- Cada objecto deverá ser responsável por uma tarefa ou objectivo específico, não lhe devendo ser atribuídas funções indevidas.
- Utilize a *keyword static* apenas quando tal se justifique e não para evitar erros do compilador.
- Recomenda-se que elabore um diagrama com as suas classes e objectos antes de iniciar o projecto, para prever a estrutura do projecto.

Tenha ainda em conta os seguintes pontos que serão importantes na avaliação:

Comente as classes, métodos e variáveis públicas segundo o formato Javadoc. Isto permitir-lhe-á gerar automaticamente uma estrutura de ficheiros HTML, descritivos do seu código, que deve incluir no seu relatório.

- Comente o restante código sempre que a leitura dos algoritmos não seja óbvia.
- Tal como sugerido acima, evite o uso abusivo de *static* e de variáveis e métodos *public*.
- Na escolha de nomes para variáveis, classes e métodos siga as convenções adoptadas na linguagem Java.
- Procure uma interface agradável com o utilizador (na obtenção de dados e disponibilização de informação). As entradas de dados por parte do utilizador deverão ser testadas e protegidas contra erros ou falhas que possam surgir.

## 4. Logística

---

- O projecto pode ser **desenvolvido individualmente** ou em **grupos (máximo 2 elementos)** (neste segundo caso surgem requisitos adicionais).
  - O projecto tem um peso de 30% (6 valores em 20) na avaliação final da disciplina.
  - O projecto tem como **prazo final de entrega o dia 22 de Junho, até às 23h00**.
  - Serão agendadas apresentações dos projectos e defesas individuais, devendo cada aluno **reservar um bloco após a submissão do projecto**. Os blocos de defesa não podem ser reservados antes da submissão do trabalho.
  - Submissão implica a entrega, em formato digital, das seguintes componentes:
    - **código da aplicação** (ficheiros com extensão `.java`, agrupados num ficheiro `.rar` ou `.zip`, incluindo ficheiros adicionais relevantes);
    - **executável** (ficheiro JAR) que permita a correcta execução da aplicação;
    - **descrição das classes**, métodos e atributos através de um **JavaDoc**;
    - **relatório final** onde seja documentada a aplicação, descrevendo-se as classes manipuladas, bem como as estruturas de dados, algoritmos ou soluções de implementação seleccionadas e respectivas justificações, com manual de utilizador.
  - Todo o material deve ser agrupado num ficheiro `.rar` ou `.zip`.
  - Os ficheiros entregues devem respeitar a nomenclatura seguinte (no caso de trabalhos em grupo devem incluir os nomes de ambos os elementos):

`LEI_PA_2025_AP3_Nome-Apelido_2025-06-xx.rar`.
  - A entrega implica a **submissão exclusivamente através da plataforma Nonio**. Nenhum trabalho será aceite fora desta plataforma ou após terminar o prazo definido (a plataforma de submissão encerrará automaticamente).
  - O desenvolvimento ocorrerá tanto em períodos de aulas como extra-aulas.
  - Sendo o foco do projecto avaliar as competências em programação do estudante, **todo o código deve ser desenvolvido exclusivamente pelo estudante**. Não são permitidas ferramentas de geração automática de código e/ou motores de Inteligência Artificial (e.g. CoPilot, ChatGPT, IDE Code Completion, funcionalidades *Drag-and-Drop*), nem código produzido por outros elementos. Projectos que incluam código não desenvolvido pelo estudante serão considerados fraude.
  - Situações de **fraude (cópia/plágio)** implicam a imediata anulação da avaliação (zero valores). A cópia de programas resultará na atribuição de nota zero a todos os alunos envolvidos. Também não é permitido copiar código da internet não referenciado.
  - Todas as fontes externas (e.g. livros, páginas de internet, recursos didáticos, ferramentas) devem ser explicitamente referenciadas, seguindo o formato indicado no modelo de relatório.
  - Os estudantes devem ter em consideração que apenas serão aceites para avaliação projectos que:
    - incluam **todos os elementos de avaliação obrigatórios**,
    - possuam **pelo menos 50% dos requisitos totalmente implementados** (das secções *Descrição Global da Aplicação e Manipulação e Armazenamento de Dados Persistente*),
    - a aplicação execute integralmente sem anomalias,
    - a aplicação servidora seja capaz de ler e escrever numa base de dados relacional, e
    - a aplicação possua uma interface gráfica funcional.
- O não cumprimento de um dos requisitos assinalados implica a anulação do trabalho.
- **O relatório final deve seguir obrigatoriamente o modelo disponibilizado.**
  - **O relatório deve conter, entre outros, os seguintes tópicos** (não de forma exclusiva, sendo recomendável a inclusão de tópicos adicionais sempre que estes permitam uma melhor clarificação do projecto e tornem o documento mais completo):
    - Descrição dos objectivos a atingir com o projecto;

- **Indicação explícita dos requisitos implementados e não implementados**, justificando os requisitos incompletos (recomenda-se o uso de uma tabela listando todos os requisitos e o seu estado);
- **Distribuição das tarefas ao longo do tempo** - indicação do planeamento das tarefas e recursos utilizados (com especial foco no tempo);
- No caso de trabalhos em grupo deve ser claro **como foi distribuído o trabalho entre os elementos do projecto**, que **responsabilidades cada elemento assumiu**;
- **Descrição das classes e respectivos métodos** (é aconselhável disponibilizar um **diagrama de classes**, e.g. VisualParadigm, UMLet, DIA Diagram Editor, Microsoft Visio);
- Descrição do **modelo de bases de dados** gerado (diagramas conceptual e físico, e.g. Sybase/SAP PowerDesigner, MySQL Workbench);
- Apresentação dos **algoritmos** adoptados;
- Indicação das **estruturas de dados** implementadas, justificando as escolhas realizadas;
- **Testes** realizados para averiguar a completude dos requisitos e o correcto funcionamento da aplicação;
- Avaliação crítica do projecto: **forças e limitações**;
- Possíveis **melhorias futuras**;
- **Referências bibliográficas** (material e fontes consultadas).
- Em anexo:
  - Manual do utilizador (descrição do funcionamento da aplicação indicando os comandos disponíveis, como executá-los e obtenção de resultados);
  - JavaDoc;
  - Documentação adicional considerada relevante para o projecto.
- A avaliação dos trabalhos terá em conta vários parâmetros, nomeadamente:
  - Programação Orientada a Objectos;
  - Usabilidade da interacção com o utilizador;
  - Manipulação de ficheiros e de estruturas de dados relacionais;
  - Completude dos requisitos;
  - Funcionalidades disponíveis;
  - Protecções implementadas, tanto na leitura de dados do utilizador, como no acesso a ficheiros;
  - Interacção com o utilizador;
  - Execução da aplicação sem erros;
  - Respeito pelas normas de codificação e documentação;
  - Qualidade do relatório;
  - Manual do utilizador;
  - Documentação do código;
  - Pesquisa e aplicação de novos conhecimentos;
  - Domínio da linguagem java
  - Apresentação e defesa do trabalho realizado.