



Projecto Final

Programação Orientada a Objectos, Estruturas de Dados e Manipulação de Ficheiros

1. Introdução

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

A aplicação tem como objectivo auxiliar o funcionamento de uma loja de informática na gestão de encomendas de equipamentos de *hardware*.

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

- **programação orientada a objectos;**
- **estruturas de dados;**
- **manipulação de ficheiros** para armazenamento de dados de forma persistente;
- bem como o desenvolvimento de **interfaces para interacção com o utilizador** (em modo texto).

2. Objectivos

Existe um conjunto de requisitos que o projecto deve respeitar e implementar (os requisitos com a etiqueta **Normal** apenas se aplicam à Época Normal da Avaliação por Exame, não sendo desenvolvidos durante a Avaliação Periódica):

2.1. Descrição global da aplicação

Gestão de acesso e utilizadores

- [R1] Permitir os utilizadores registarem-se e autenticarem-se na aplicação.
- [R2] Permitir o acesso à aplicação por 3 tipos de utilizadores: administradores, técnicos e clientes.
- [R3] Os **utilizadores** são caracterizados pelos atributos login, password, nome, estado (activo/inactivo), email e tipo (administradores, técnicos ou clientes).
- [R4] Cada utilizador apenas pode alterar a sua própria informação, não podendo alterar ou visualizar dados de outros utilizadores, ou criar utilizadores.
- [R5] O login e email devem ser únicos.
- [R6] Os **clientes** e **técnicos** caracterizam-se adicionalmente por número de identificação fiscal (NIF), morada e contacto telefónico.
O NIF e contacto telefónico são únicos.
- [R7] Caso não existam utilizadores criados, a aplicação deve solicitar a criação de um utilizador (administrador).
- [R8] Após a autenticação, a aplicação deve apresentar a mensagem
"Bem-vindo [*nome utilizador*]".

[R9] Quando a aplicação estiver a encerrar, deve apresentar a mensagem

“Adeus [*nome utilizador*]”.

Funções

[R10] Os **administradores** aprovam os pedidos de registo dos utilizadores. Todos os pedidos devem ser aprovados antes de poderem ser usados para autenticação.

[R11] Os **administradores** aprovam e encerram os pedidos de serviço (encomenda), bem como associam um técnico responsável a cada serviço.

[R12] Os **técnicos** introduzem equipamentos, categorias e fornecedores.

[R13] Os **clientes** realizam pedidos de compras de equipamentos, que devem ser aprovadas pelos técnicos.

[R14] Os **administradores** podem gerir todas as vendas realizadas.

[R15] Os **técnicos** apenas podem gerir as suas vendas.

[R16] Os **clientes** apenas podem gerir as suas compras.

Acções

[R17] O serviço de compra inicia com o **cliente** a solicitar uma encomenda, indicando os produtos pretendidos.

[R18] O **administrador** aprova o serviço e associa um técnico responsável.

[R19] Após ser designado para um serviço, o **técnico responsável** regista a informação do processo, nomeadamente pode criar sub-tarefas e associar outros técnicos adicionais às sub-tarefas, e gere o processo até à sua conclusão.

[R20] Um **serviço de compra** (encomenda), além de ter associado um técnico responsável, inclui a listagem dos produtos incluídos (equipamentos de *hardware*, podendo definir a quantidade), uma data de realização do pedido, uma descrição, o estado e o tempo que demorou a ser processado o valor total da venda.

[R21] Adicionalmente, cada serviço possui um código identificador único e sequencial.

[R22] O estado do serviço apresenta um dos seguintes valores:

- a) se o serviço está **submetido** (cliente realizou o pedido, mas ainda não foi aceite pelo administrador);
- b) se a serviço foi **aceite** (administrador aceita o serviço);
- c) ou se já foi **concluído** (técnico responsável termina o processo).

[R23] Quando um técnico conclui um serviço deve indicar o tempo despendido no processamento do pedido.

O valor total da venda deve ser calculado automaticamente (somatório do custo dos vários equipamentos e respectivas quantidades).

[R24] A aplicação deve permitir introduzir e gerir equipamentos de *hardware*.

Os **equipamentos de hardware** são caracterizados por uma marca, modelo, código interno, série, versão, voltagem, quantidade em stock, preço de venda, observações e se o equipamento é OEM (*Original Equipment Manufacturer*).

[R25] O código de um equipamento deve ser único.

[R26] Cada **equipamento** deve ter associado uma lista até 6 fornecedores, podendo ser introduzido um novo fornecedor a qualquer momento.

[R27] Cada **fornecedor** é caracterizado por nome, morada e contacto telefónico.

[R28] Cada **equipamento** deve apresentar uma lista até 4 categorias, podendo ser introduzida uma nova categoria a qualquer momento.

[R29] Cada **categoria** é caracterizada pela sua designação e família (atributos alfanuméricos).

[R30] Cada **série** possui uma geração e sequência (atributos numéricos).

[R31] Cada **versão** é caracterizada pela unidade, valor alfa e valor beta (atributos numéricos).

[R32] É possível alterar a quantidade em stock de um **equipamento**.

- [R33] Sempre que um equipamento é vendido, a quantidade em *stock* deve ser actualizada.
- [R34] (**Recurso**) Deve ser emitido um recibo ao cliente, indicando o NIF e custo total do serviço realizado.
- [R35] (**Recurso**) Deve existir um módulo de estatísticas que indica o número e valor total de serviços realizados por um cliente.

Listagens e pesquisas

- [R36] Deve ser possível ordenar **utilizadores** por ordem alfabética do nome.
- [R37] Deve ser possível ordenar **equipamentos** por designação.
- [R38] (**Normal**) Deve ser possível ordenar **serviços** por ordem da data (do mais recente ao mais antigo).

- [R39] Deve ser possível listar todos os **utilizadores**.
- [R40] Deve ser possível listar **utilizadores** por tipo.
- [R41] Deve ser possível listar todos os **serviços**.
- [R42] Deve ser possível listar todos os **serviços** associados a um cliente.
- [R43] Deve ser possível listar todos os **serviços** por estado.
- [R44] Deve ser possível listar **serviços** com tempo despendido superior a um determinado limite (introduzido pelo utilizador no momento de pesquisa).
- [R45] Deve ser possível listar todos os **equipamentos**.
- [R46] Deve ser possível listar **equipamentos** que sejam (ou não) OEM.
- [R47] Deve ser possível listar **equipamentos** com uma quantidade de stock abaixo de um determinado limite (introduzido pelo utilizador no momento de pesquisa).
- [R48] (**Normal**) Deve ser possível listar os **equipamentos** mais adquiridos (e.g. 10 equipamentos mais adquiridos).

- [R49] Deve ser possível pesquisar **utilizadores** por login ou nome.
- [R50] Deve ser possível pesquisar **serviços** por código ou uma palavra que surja na descrição.
- [R51] Deve ser possível pesquisar **equipamentos** por marca ou código.
- [R52] Deve ser possível pesquisar **equipamentos** que possuem uma determinada categoria.

- [R53] Deve ser possível realizar pesquisas avançadas, ou seja, apresentar todos os registos que apresentem um termo de pesquisa, mesmo que parcialmente

(e.g. termo de pesquisa “Ana” deve apresentar como resultado “Ana Sousa”, “Ana Silva” e “Anabela”).

- [R54] Os **clientes** podem listar e pesquisar os seus serviços que realizaram.
- [R55] Os **técnicos** podem listar e pesquisar os seus serviços que processaram.
- [R56] Os **administradores** podem listar e pesquisar todos os serviços.

2.2. Manipulação e armazenamento de dados persistente

- [R57] O acesso à aplicação deve ser restringido com credenciais (*login/password*), informação que deverá ser armazenada num ficheiro de texto “credenciais_acesso.txt”.
- [R58] Durante o encerramento da aplicação, os dados devem ser automaticamente guardados num ficheiro de objectos “dados_apl.dat”.
- [R59] A aplicação, no arranque, deve automaticamente ler os dados do ficheiro de objectos “dados_apl.dat”, caso este exista, e informar o utilizador que os dados foram lidos com sucesso.

Caso o ficheiro esteja indisponível deve surgir uma mensagem a informar a situação.

- [R60] (Normal) Deve ser possível exportar todos os serviços realizados para um ficheiro de texto em formato CSV (*Comma Separated Values*), com a estrutura
- <data>, <valor>, <cliente>, <produtos>
- ordenados do serviço mais recente para o mais antigo.

2.3. Gestão geral da aplicação

Interacção com o utilizador

- [R61] Disponibilizar uma interface em modo texto onde o utilizador possa interagir e controlar a aplicação.

Monitorização de acessos

- [R62] Todas as acções dos utilizadores deverão ser guardadas num ficheiro de texto denominado "log.txt".
- Este ficheiro deverá ser escrito de forma sequencial, apresentando as acções mais recentes no início do ficheiro.
- As entradas deverão ter o seguinte formato: <utilizador> <acção>.
- [R63] Deve ser possível consultar o *log* de acções através da aplicação.
- [R64] Deverá ser registado o número total (até ao momento) de execuções do sistema e o *login* do último utilizador que acedeu à aplicação.
- Esta informação deverá ser guardada num ficheiro de objectos e actualizada sempre que a aplicação é executada, apresentando os dados ao utilizador.
- Quando a aplicação é terminada esta informação deve ser novamente armazenada no ficheiro, sendo este denominado "info_sistema.dat".

Programação Orientada a Objectos

- [R65] A aplicação deve estar correctamente estruturada, tendo em conta o paradigma Orientado a Objectos e recorrendo à linguagem Java.
- [R66] Implemente as estruturas de armazenamento necessárias, procurando otimizar os recursos utilizados.

Validação de dados e notificações

- [R67] Valide todas as leituras de dados do utilizador (e.g. verifique se os nomes são únicos).
- [R68] Sempre que necessário, apresentar ao utilizador mensagens informativas adequadas.
- Quando um utilizador realizar uma acção sobre a aplicação, esta deve informar se acção foi realizada com sucesso ou insucesso.

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 (não é obrigatório gerar Javadoc). Comente o restante código sempre que a leitura dos algoritmos não seja óbvia
- 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 grupo (máximo 2 elementos)** na Avaliação Periódica. Na Época Normal da Avaliação por Exame, o projecto deve ser desenvolvido individualmente.
- O projecto tem um peso de 50% (10 valores em 20) na avaliação final da unidade curricular.
- O projecto tem como **prazo final de entrega o dia 22 de Janeiro, até às 12h00**.
- Será submetido um **relatório intermédio até ao dia 14 de Dezembro, 23h00** (apenas para Avaliação Periódica, não aplicável à Avaliação por Exame, Época Normal).
- Serão agendadas apresentações dos projectos e defesas individuais, devendo cada estudante/grupo **reservar um bloco após a submissão final do projecto**. Os blocos de defesa não podem ser reservados antes da submissão do trabalho.
- Submissão final implica a entrega, em formato digital, de três componentes:
 - **código da aplicação** (ficheiros com extensão `.java`, incluindo ficheiros adicionais relevantes);
 - **relatório final** seguindo o modelo disponibilizado, 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;
 - **manual do utilizador**, instruindo o utilizador de como poderá interagir com a aplicação.
- Todo o material deve ser agrupado num ficheiro `.rar` ou `.zip`.
- O ficheiro do projecto submetido deve respeitar a nomenclatura seguinte:

LEI_Prog_2024_AP3_Nome-Apelido_2025-01-xx.rar (Avaliação Periódica), ou

LEI_Prog_2024_AE-EN_Nome-Apelido_2025-01-xx.rar (Época Normal da Avaliação Exame).
- 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 num período 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.g. CoPilot, ChatGPT, IDE Code Completion), nem código produzido por outros elementos. Projectos que incluam código não desenvolvido pelo estudante serão considerados plágio.
- 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**, e
 - **a aplicação seja capaz de ler e escrever em todos os ficheiros de texto e de objectos** definidos no enunciado.
- 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, referindo o tempo gasto em cada requisito). Em caso de trabalhos conjuntos explicitamente referir que requisitos foram realizados por cada elemento e qual o tempo dedicado;
 - **Descrição das classes e respectivos métodos**;
 - 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: **pontos fortes e pontos fracos**;
 - 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);
 - Documentação adicional considerada relevante para o projecto.
- O código e execução da aplicação apresenta um peso de 85%, enquanto a documentação (relatório e manual) apresenta um peso de 15%.
- 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;
 - Tratamento de excepções;
 - Execução da aplicação sem erros;
 - Respeito pelas normas de codificação;
 - 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.