



Projecto

Programação Orientada a Objectos, Estruturas de Dados, Manipulação de Ficheiros, Acesso a Bases de Dados Relacionais

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 e com suporte para bases de dados relacionais.

A aplicação tem como objectivo auxiliar uma editora no processo de revisão de obras (livros). O sistema deve permitir armazenar toda a informação inerente ao processo e possibilitar a consulta e pesquisa da mesma.

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 texto).

2. Objectivos

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

2.1. Descrição global da aplicação

Gestão de acesso e utilizadores

- [R1] Permitir aos utilizadores registarem-se e autenticarem-se na aplicação.
- [R2] Permitir o acesso à aplicação por 3 tipos de utilizadores: gestores da plataforma (ou administradores), autores e revisores.
- [R3] Os **utilizadores** são caracterizados pelos atributos nome, login, password, estado (activo/inactivo), email e tipo (gestor, autor ou revisor).
- [R4] Tanto o atributo login como o email devem ser únicos.
- [R5] O email deve apresentar um formato válido ([designação] @ [entidade] . [domínio]).
- [R6] Cada **utilizador** apenas pode alterar a sua própria informação, não podendo alterar ou visualizar dados de outros utilizadores, ou criar novos utilizadores.
- [R7] A excepção ao requisito anterior são os **gestores**, que podem visualizar e alterar dados de todos os utilizadores e criar novos utilizadores, nomeadamente outros gestores.

[R8] Tanto os **autores** como os **revisores** têm adicionalmente os parâmetros número de contribuinte, contacto telefónico e morada.

Os **autores** possuem ainda estilo literário (e.g. drama, ficção, thriller) e data de início de actividade.

Os **revisores** possuem, adicionalmente, área de especialização (e.g. ciências, literatura, artes) e formação académica.

[R9] O atributo número de contribuinte deve ser único e possuir 9 dígitos.

[R10] O atributo contacto telefónico deve possuir 9 dígitos e iniciar pelos dígitos 9, 2 ou 3.

[R11] Os **autores** e **revisores** devem registar-se na plataforma.

Os **gestores** são criados por outros gestores.

[R12] Todos os pedidos de registo de novos utilizadores devem ser notificados aos **gestores** através da aplicação.

[R13] Os **gestores** aprovam ou rejeitam os pedidos de registo dos utilizadores.

Todos os pedidos devem ser aprovados antes de poderem ser usados para autenticação.

[R14] Caso uma conta seja reprovada pelo gestor ou ainda não tenha sido analisada, quando o **utilizador** tentar usar essas credenciais deve surgir uma mensagem informativa.

[R15] Em qualquer momento um **gestor** pode inactivar ou activar uma conta de um utilizador.

[R16] Um **utilizador** pode solicitar que a sua conta seja removida do sistema. Neste caso deve surgir uma notificação aos gestores que podem aceitar ou recusar o pedido.

Se o pedido for aceite, a informação pessoal do utilizador deve ser omitida do sistema, embora os registos associados ao utilizador devam ser mantidos.

[R17] Caso não existiam utilizadores criados na primeira execução, a aplicação deve solicitar credenciais para criar uma conta gestor.

[R18] Após a autenticação, a aplicação deve apresentar a mensagem

“Bem-vindo [*nome utilizador*]”.

[R19] Quando a aplicação estiver a encerrar, deve apresentar a mensagem

“Adeus [*nome utilizador*]”.

Acções

(Intervenientes e responsabilidades)

[R20] Os **revisores** gerem (aprovam e finalizam) o processo de revisão.

[R21] Os **autores** submetem obras para revisão, que devem ser revistas por um ou mais revisores, e realizam o respectivo pagamento.

[R22] Os **revisores** são responsáveis pela realização da revisão.

[R23] Cada **obra** para revisão é caracterizada pelo autor, título, sub-título (opcional), estilo literário (e.g. drama, ficção, *thriller*), tipo de publicação (e.g. capa dura, de bolso) número de páginas, número de palavras, código ISBN, número de edição, data de submissão e data de aprovação.

O código ISBN deve possuir um valor único.

[R24] Uma **revisão** está associada a uma obra, autor, gestor e revisor responsável.

[R25] Uma revisão é caracterizada por um número de série, que é composto por um número sequencial (a cada pedido o número incrementa), seguido da data no formato AAAAMMDDHHMMSS.

Por exemplo, se já ocorreram 95 pedidos até ao momento, e às 10h20m10s do dia 21 de Fevereiro de 2023 surge um novo pedido, o mesmo terá o número 96**20230221102010**.

[R26] Adicionalmente, uma revisão inclui a data da realização, o tempo decorrido (calculado automaticamente desde o início até ao fim da revisão) uma listagem de anotações realizadas, observações genéricas inseridas pelo revisor, um custo do processo e o seu estado.

[R27] As **anotações** são caracterizadas por uma descrição, a página e o parágrafo onde ocorrem e a data em que foram realizadas.

(Processo)

- [R28] O processo inicia com o **autor**, que solicita uma revisão para uma das suas obras.
Cada pedido recebe um número ISBN aleatório entre 1 e 1 000 000.
- [R29] O pedido de revisão tem de ser aprovado pelo **gestor**. O pedido pode ser aceite ou rejeitado.
Se aceitar o pedido, o gestor deve atribuir um ou mais revisores. Deve sempre existir um revisor responsável.
Se o pedido for rejeitado o autor deve ser notificado e o processo arquivado.
- [R30] Quando um processo de revisão é atribuído a um **revisor** este deve ser notificado.
O revisor pode aceitar ou rejeitar o processo.
- [R31] Caso um revisor rejeite um processo, o gestor responsável pelo processo deve ser notificado e seleccionar outro revisor.
Na nova listagem de possíveis revisores a atribuir, não deve surgir o(s) revisor(s) que rejeitou (rejeitaram) anteriormente o pedido.
- [R32] Após aceitar o processo, o **revisor** inicia a revisão. Quando concluir a tarefa, encerra essa revisão.
A revisão não tem de ter terminada numa única sessão da aplicação.
- [R33] No final o **autor** realiza o pagamento do processo e o gestor será notificado.
- [R34] Após a confirmação do autor, o **gestor** arquiva o processo.
- [R35] Tanto o autor como o gestor podem consultar o estado de uma revisão:
a) *iniciada* (**autor** submeteu pedido, a aguardar autorização do gestor),
b) *aceite* (**gestor** aceitou o pedido, em espera de execução);
c) *decorrer* (**revisor** iniciou a revisão);
d) *finalizada* (**revisor** terminou a revisão);
e) *arquivado* (o **gestor** encerra o processo após o **autor** pagar o processo).
- [R36] Cada **autor** pode possuir diversas obras e cada obra só está associada a um autor.
- [R37] Sempre que uma nova obra é inserida no sistema deve-se verificar se o título já existe, bloqueando a sua inserção em caso de duplicação.
- [R38] Uma obra pode possuir várias revisões.
- [R39] Para garantir que as obras não contêm plágio, são utilizadas aplicações específicas. Estas aplicações possuem um número limitado de licenças, com prazo de utilização (usualmente um ano).
Cada revisão pode usar uma ou mais licenças.
- [R40] O gestor é responsável pela inserção de licenças no sistema e actualização do seu número.

Notificações

- [R41] Quando um novo utilizador se registou na plataforma o **gestor** deve ser notificado.
- [R42] Sempre que existe um pedido de revisão o **gestor** deve ser notificado.
- [R43] Se um pedido de revisão for rejeitado o **autor** deve ser notificado.
- [R44] Quando um pedido de revisão é atribuído a um **revisor** este deve ser notificado.
- [R45] Se um revisor rejeitar um pedido, o **gestor** deve ser notificado.
- [R46] O **gestor** deve ser notificado quando um processo de revisão for confirmado/pago pelo autor.
- [R47] Sempre que um processo de revisão passou 10 dias sem ser finalizado o **gestor** deve ser notificado.

Listagens e pesquisas

- [R48] Todas as listagens devem permitir visualizar os resultados por páginas de **10 registos** cada.

- [R49] Todas as listagens devem permitir ordenar os resultados, no sentido **ascendente ou descendente**.
- [R50] Todas as listagens devem permitir realizar pesquisas para filtrar os resultados.
- [R51] Os **gestores** podem listar todos os utilizadores, ordenando por nome.
- [R52] Os **gestores** podem pesquisar utilizadores por nome, *login* ou tipo.
- [R53] Os **gestores** podem listar todas os pedidos de revisão ordenados por data de criação, título de obra ou por autor.
- [R54] Os **gestores** podem listar todas os pedidos de revisão ainda não finalizados, ordenados por data.
- [R55] Os **gestores** podem pesquisar pedidos de revisão por identificador, estado ou autor.
- [R56] Os **gestores** podem pesquisar pedidos de revisão submetidos dentro de um intervalo temporal (período entre duas datas a indicar pelo utilizador).
- [R57] Os **gestores** podem listar os processos de revisão de qualquer obra a partir do título.
- [R58] Os **autores** podem listar os seus pedidos de revisão ordenando por data de criação ou por número de série.
- [R59] Os **autores** podem pesquisar os seus pedidos de revisão por data de criação, por título ou estado.
- [R60] Os **autores** podem listar as suas obras ordenadas por data de submissão ou por título.
- [R61] Os **autores** podem pesquisar os seus obras por data de registo ou matrícula.
- [R62] Os **revisores** podem listar os seus pedidos de revisão ordenando por data de criação ou por título.
- [R63] Os **revisores** podem pesquisar os seus pedidos de revisão por data de criação, por título ou estado.
- [R64] Devem ser implementados mecanismos de pesquisa avançada, 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”).

2.2. Manipulação e gestão de bases de dados relacionais

- [R65] Acesso a uma **base de dados relacional** que permita gerir toda a informação necessária para a execução da aplicação.
- [R66] O acesso à aplicação deve ser restringido com credenciais (*login/password*), informação que deverá ser armazenada numa base de dados relacional.
- [R67] Os parâmetros de acesso à base de dados (IP, porto, nome da base de dados, *login* e *password*) devem ser armazenados num **ficheiro de texto** (*Properties*), sendo disponibilizada uma interface que possibilite a sua alteração antes do arranque da aplicação (antes do processo de autenticação).
- Estes parâmetros não necessitam de ser definidos em todas as execuções da aplicação, apenas quando o utilizador solicitar.

2.3. Gestão geral da aplicação

Interacção com o utilizador

- [R68] Disponibilizar uma **interface em modo texto** onde o utilizador possa interagir e controlar a aplicação.
- A organização da interface e escolha de componentes fica ao critério do aluno. Soluções eficientes de menus, respeitando as regras de interacção com o utilizador serão bonificadas.
- [R69] Calcular o tempo que a aplicação demora a executar a aplicação, desde o arranque até ao utilizador seleccionar a opção de saída, apresentando, no final do processo, a seguinte informação:

Início do processo: Terça-Feira; 2023-03-01 11:21:11

Fim do processo: Terça-Feira; 2023-03-01 11:21:56

Tempo de execução: 45132 Milissegundos (45 Segundos; 0 Minutos; 0 Horas)

Monitorização de acessos

[R70] Deverá ser mantida a informação genérica do sistema, como por exemplo a data da última execução da aplicação e o número total (até ao momento) de execuções do sistema.

[R71] Deve existir um **registo de acções (log)** dos utilizadores no formato:

<data> <hora> <utilizador> <acção>.

O *log* deve ser armazenado numa base de dados relacional.

[R72] Deve ser possível listar o conteúdo do *log* através da aplicação.

[R73] Deve ser possível pesquisar registos por utilizador no *log*.

Programação Orientada a Objectos

[R74] A aplicação deve estar correctamente estruturada, tendo em conta o paradigma Orientado a Objectos, recorrendo à linguagem Java.

[R75] Implemente as estruturas de armazenamento necessárias, procurando optimizar os recursos utilizados.

Validação de dados e notificações

[R76] Valide todas as leituras de dados do utilizador (e.g. verifique se os nomes são únicos).

[R77] 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 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 grupo (máximo 2 elementos)**.
- 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 18 de Março, até às 23h00**.
- Serão agendadas apresentações dos projectos e defesas individuais, devendo cada estudante/grupo **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, de quatro 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`.
- O ficheiro do projecto submetido deve respeitar a nomenclatura seguinte:
`LEI_PA_2024_AP1_Nome-Apelido_2024-03-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.
- 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.
- 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 numa base de dados relacional**.
- 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);
 - **Descrição das classes e respectivos métodos** (é aconselhável disponibilizar um **diagrama de classes**, e.g. VisualParadigm, UMLet, DIA Diagram Editor, Microsoft Visio);
 - Apresentação dos **algoritmos** adoptados;
 - Indicação das **estruturas de dados** implementadas, justificando as escolhas realizadas;

- Descrição do **modelo de bases de dados** gerado (diagramas conceptual e físico, e.g. Sybase/SAP PowerDesigner, MySQL Workbench);
 - **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);
 - **JavaDoc**;
 - Documentação adicional considerada relevante para o projecto.
- O código e execução da aplicação apresenta um peso de 85%, enquanto a documentação (relatório, manual e JavaDoc) 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;
 - 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.