

Paradigmas de Programação

Projeto de Grupo 2018/2019

Existem várias aplicações online gratuitas que permitem fazer a gestão de projetos, com algumas limitações. O Freedcamp (<https://freedcamp.com/>) e Trello (<https://trello.com/>) são exemplos deste tipo de aplicações. Nesta edição da unidade curricular de Paradigmas de Programação pretende-se a criação de uma aplicação informática cujo funcionamento se baseie numa dessas aplicações e que implemente as funcionalidades indicadas neste documento. A aplicação deverá ser desenhada de acordo com os princípios do Paradigma Orientado a Objetos e utilizar o Java SE.

A aplicação criada será uma versão simplificada, focando-se na gestão de tarefas. Os utilizadores têm de se registar para terem acesso às suas funcionalidades. Os utilizadores autenticados têm a possibilidade de criarem projetos e fazerem a sua gestão, incluindo criarem listas de tarefas e associarem pessoas aos projetos. Cada projeto tem um owner, que é o utilizador que criou o projeto, e só ele pode alterar os dados gerais do projeto, como o título, a descrição, a data de início e de fim do projeto. O owner do projeto pode, em cada momento, associar pessoas (outros utilizadores registados) ao projeto. Todos os utilizadores registados têm acesso à informação dos seus projetos (de que é owner) e dos projetos a que está associado, podendo, ainda, para cada projeto, criar listas de tarefas, e, para cada lista, criar tarefas. Para cada tarefa é indicado: título, descrição, associar pessoa à tarefa (um dos utilizadores associados ao projeto), prioridade (baixa, média, alta), data de fim, estado da tarefa (não iniciada, em progresso, terminada). Para cada lista ou tarefa deve ficar registado quem a criou.

Funcionalidades

A aplicação informática deverá ser composta por duas partes: a parte de back-end que suporta o funcionamento do sistema, e a parte de front-end que suporta a interface gráfica que permite a interação com o utilizador.

O back-end e o front-end deverão estar implementados em pacotes distintos, ou seja, todas as classes que digam respeito à interface gráfica deverão estar arrumadas num pacote e as classes respeitantes ao negócio, noutro.

A entrada e saída de dados deve ser realizada via interface gráfico, assente numa ou mais janelas. O desenvolvimento da interface deve ser baseado no uso das classes existentes nos packages AWT/Swing do JAVA. Pretende-se ainda que a aplicação tenha capacidade de persistência, ou seja, que tenha capacidade de guardar em ficheiro o estado do sistema num determinado momento e recuperá-lo quando pretendido. Refere-se que não é permitida a utilização de bases de dados. Ainda em relação à entrada e saída de dados importa referir que se pretende que o sistema tenha níveis de robustez adequados, ou seja, que seja capaz de se comportar adequadamente em situações de erro (nomeadamente aquando da introdução de dados por parte dos utilizadores).

A seguir estão listadas as funcionalidades que deverão ser implementadas:

1. Registrar
2. Autenticar
3. Alterar dados do seu perfil

4. Criar projetos
5. Alterar dados do projeto de que é owner
6. Associar pessoas (outros utilizadores) aos projetos de que é owner
7. Para cada projeto, criar listas de tarefas
8. Para cada lista de tarefa, criar tarefas
9. Alterar qualquer tipo de informação respeitante às listas e às tarefas
10. Alterar o estado das tarefas. Quando a tarefa é indicada como terminada, é registada a data em que a tarefa foi concluída.
11. Visualizar a lista de projetos do utilizador, indicando o título, data de início e a data de fim do projeto
12. Selecionar um projeto para edição
13. Listar todas as tarefas do projeto (título, prioridade, data de fim, estado, utilizador associado)
14. Filtrar a listagem de tarefas por: utilizador que criou; utilizador associado; prioridade; estado; data de fim.
15. Ordenar (ascendente ou descendente) a listagem de tarefas por: ordem alfabética do título; data de fim.
16. Listar as tarefas que estão atrasadas
17. Visualizar um dashboard geral dos projetos que apresente:
 - a. o número e percentagem de projetos concluídos, em curso e atrasados;
 - b. os 3 projetos mais atrasados;
 - c. o projeto mais problemático, isto é, mais pessoas envolvidas e mais tarefas concluídas.

Valorização

A aplicação dos conceitos e dos princípios fundamentais subjacentes ao paradigma de programação orientada aos objetos será amplamente valorizada.

A classificação dos trabalhos terá ainda em consideração aspetos como:

- Qualidade da execução do programa (com ou sem erros, validações, testes, ...)
- Qualidade da satisfação dos requisitos
- Facilidade de utilização do sistema
- Adequação da conceção e organização das classes
- Qualidade do código fonte apresentado
 - Correção (avaliação se o código está coerente com a especificação do enunciado e se não tem erros)
 - Design (avalia a forma como o código está escrito (eficiente, elegante, reutilizável))
 - Estilo (avalia a legibilidade do código (comentado, indentado, nomes de variáveis e funções adequadas,...))
- Desempenho do grupo no decorrer da apresentação

Datas e Formalidades de Entrega

O trabalho deve ser realizado em **grupos de 3 estudantes inscritos no mesmo turno PL**. O projeto, em formato zip, deve ser submetido no BlackBoard até **26 de maio de 2019**. Juntamente com o projeto deve ainda ser entregue um relatório básico (em formato pdf) incluindo uma tabela contendo todas as funcionalidades implementadas e não implementadas, o diagrama de classes e a documentação gerada (*javadoc*) relativa às classes criadas.