

Programação Orientada a Objetos

Trabalho Prático

Introdução

Nas micro e pequenas empresas prestadoras de serviços os sistemas integrados de gestão contribuem para um melhor desempenho, através do aumento da eficiência no funcionamento e pela melhoria da eficácia na prestação desses mesmos serviços. As soluções informáticas procuram abranger o maior número possível de diferentes contextos de negócio, num esforço de síntese mas também numa natural procura de hegemonia de mercado. Contudo, verifica-se que as diferentes realidades condicionam significativamente o código.

É comum às mais diversas soluções informáticas a existência de um módulo dedicado à gestão do pessoal da própria organização. Contudo, quanto maior a diversidade das atividades dos funcionários maior a complexidade da solução informática, seja pela diferença de tarefas ou pela especificidade das formações individuais. Apesar do objetivo ser a gestão do pessoal e isto motivar a existência de um módulo dedicado, é bem possível a existência de módulos com características diferentes consoante a organização onde está implementado.

Para a gestão dos recursos próprios da organização também se verifica a necessidade de atender às especificidades das tarefas por parte das soluções informáticas: numa farmácia as condições de funcionamento e de armazenamento dos materiais são diferentes das necessárias numa oficina de mecânica automóvel. Os correspondentes custos de funcionamento são, naturalmente, também diferentes. Mas a gestão dos recursos próprios da organização, com os respetivos custos, impõe normalmente um módulo dedicado a este efeito.

Outros módulos também são frequentes: gestão de *stocks*, contabilidade, controlo orçamental, gestão da produção, gestão comercial, etc.

Objetivo

Tendo como referência uma farmácia, modelizar um sistema de informação para este tipo de empresa que seja constituído por, pelo menos, quatro módulos:

- gestão de pessoal;
- gestão de *stocks*;
- contabilidade;
- gestão de recursos.

Para cada módulo é necessário um conjunto básico de funcionalidades. Não é exigível aos alunos conhecimentos especializados em recursos humanos ou contabilidade, por exemplo. No caso da gestão de pessoal será possível, no mínimo, a criação de um conjunto de dados pessoais para cada novo empregado, a atualização destes dados e a eliminação da lista de ativos mantendo o registo histórico. Considerar, também, um mecanismo de registo do início e fim da atividade diária de cada empregado.

O detalhe funcional de cada módulo constará na descrição dos requisitos funcionais expressos no relatório que acompanha cada trabalho. Por outras palavras, cada módulo estará enquadrado por uma descrição dos requisitos funcionais a cumprir. Realço que os requisitos funcionais resultam do interesse do cliente e das observações do técnico. Quando estabelecidos, são para cumprir.

A solução a desenvolver apresenta um interface gráfico ao utilizador, possibilitando o acesso aos demais aspetos funcionais do sistema. A interface gráfica não deve “esconder” funcionalidades desenvolvidas, assim como não deve apresentar possibilidades funcionais que...não funcionam. A interface gráfica tornará, inevitavelmente, evidente o estado de complexidade ou aprofundamento da arquitetura do sistema face aos objetivos do mesmo. A interface será desenvolvida com base no *package swing*, e implementada corretamente em programação orientada a objetos.

O sistema fará uso de registos em ficheiros CSV (*Comma Separated Values*).

Todo o código é acompanhado por ficheiros padronizados e resultantes de *javadoc*.

Todo o código é acompanhado pela estrutura UML, que constará no relatório.

O relatório é elaborado em conformidade com o estabelecido da aula de apresentação (de acordo com o Anexo I do Regulamento Geral de Mestrados e Doutoramentos), conforme indicado na aula de apresentação.

À exceção das palavras padronizadas da *java*, todo o código aplicará a língua portuguesa (campos, classes, métodos, pacotes, comentários), **não sendo considerado código que não cumpra esta condição.**

Cada trabalho é desenvolvido por um grupo com o máximo de três alunos.

Os prazos e datas de entrega dos trabalhos são os que constam na aula de apresentação de POO. Eventuais ajustes poderão surgir e serão divulgados oportunamente.

Avaliação

Se bem que a especificidade funcional de cada módulo não é critério prioritário na avaliação, são os aspetos técnicos de implementação da solução informática ou de desenvolvimento do código que constituem prioridade na avaliação:

- código desenvolvido sob o paradigma de objetos (java);
- modularidade;
- escalabilidade;
- compreensão (comentários).

As boas práticas na programação estão diretamente relacionadas com a facilidade de manutenção do código e vice-versa. A modularidade, escalabilidade e facilidade de compreensão do código estão entre os fatores que mais contribuem para o seu fácil entendimento.

A par das boas práticas em *software* também será uma arquitetura mais elaborada, com mais e coerentes funcionalidades que contribuirão para melhor classificação.

Também a par das boas práticas de *software* e da arquitetura do sistema é a profundidade de conhecimento da sintaxe da *java* conjuntamente com o paradigma de desenvolvimento de código por objetos (encapsulamento, herança, polimorfismo) que constituem critério prioritário na avaliação.

Classificação:

Trabalhos não compilam – 0 a 5 valores;

Trabalhos compilam:

- comentários e relatório deficientes – 5 – 12;
- resultados corretos, qualidade do código, qualidade dos comentários, qualidade da arquitetura e qualidade do relatório 6 – 20.

O professor,

Raul Dionísio