

# Enunciado Trabalho Final (2017/2)

## A) Objetivo:

O objetivo deste trabalho é consolidar o conhecimento sobre conceitos e construção de sistemas empresariais orientados a objetos em arquiteturas multicamadas através da exploração dos tópicos discutidos na disciplina de Fundamentos de Desenvolvimento de Software.

## B) Enunciado do problema:

Estamos interessados em um sistema de informação para o controle de leilões.

Os leilões suportados pelo sistema são do tipo:

- Leilão de demanda. Compradores fazem lance de demanda para comprar um lote que o vendedor pretende disponibilizar pelo maior preço. Vence o participante que fez o maior lance de demanda primeiro entre todos os lances, desde que seu lance seja maior do que o preço mínimo estabelecido pelo vendedor.

O sistema deve suportar a realização de lances da seguinte forma:

- Aberto. A qualquer momento é possível verificar o valor de todos os lances já gerados e qual o lance vencedor no momento.

Ao ser criado um leilão no sistema, todo leilão deve possuir a identificação do usuário vendedor. Por fim, todo leilão possui um preço mínimo associado ao lote.

Lote é um conjunto formado de um ou mais bens que serão leiloados. Os bens a serem leiloados devem possuir uma descrição breve, uma descrição completa e uma categoria (por exemplo, mobiliário, informática, etc).

Sobre os lances, é necessário conhecer o valor do lance e identificador de qual usuário realizou o lance.

Sobre os usuários do sistema, é necessário conhecer pelo menos nome, cpf ou cnpj e endereço de e-mail.

Ao final do período que um determinado leilão está aberto, o sistema deve determinar o vencedor do leilão.

Em suma, o sistema deve permitir os seguintes casos de uso:

- Cadastrar usuários;
- Cadastrar leilão e bens a serem leiloados;
- Encerrar leilão;
- Visualizar detalhes de um leilão já terminado ou em andamento;
- Cadastrar lance em um leilão em andamento;
- Cancelar lance em um leilão em andamento.

## C) Base de dados:

O trabalho deve fazer uso de uma base de dados. Pode-se o Derby (como testado em aula) ou outro gerenciador de banco de dados.

## D) Requisitos:

Os seguintes itens são obrigatórios na implementação do sistema:

- Arquitetura multicamada (pelo menos 3).
- Uso dos padrões de projeto explorados em sala de aula, sendo obrigatoriamente:
  - Uso do padrão "Facade" para isolar a camada de domínio da camada de apresentação;
  - Uso do padrão arquitetural "Domain Model" na camada de domínio;
  - Uso do padrão "DAO" na camada de persistência.

- Interface gráfica de usuário desktop (interface textual de console não será aceita);
- Persistência em banco de dados relacional;
- A camada de persistência deve ser implementada sem a utilização de frameworks mapeadores objeto/relacional (como JPA, Hibernate, etc);
- O banco de dados deverá ter sido previamente populado (um arquivo contendo os scripts para geração do BD devem ser entregues juntamente com o código fonte) com, no mínimo, os valores necessários para uma boa cobertura de casos de teste.

#### **E) Desenvolvimento e avaliação do trabalho:**

- O trabalho pode ser realizado individualmente ou em grupos de, no máximo, 3 alunos.
- É tarefa do grupo garantir que o sistema esteja apto a ser executado quando entregue (fontes, executáveis, libs e banco de dados populados devem ser entregues).
- Mensagens de erro apresentadas durante a execução do programa, mesmo que a aplicação não pare de executar, serão consideradas como erros de execução.
- Os trabalhos serão avaliados de acordo com o que é pedido no enunciado e o que foi realizado com sucesso pelo sistema. Também serão avaliadas a modelagem do sistema (correta criação das classes necessárias, com seus atributos e métodos, encapsulamento, e correto estabelecimento de relações entre as classes) e sua implementação de acordo com os conceitos de orientação a objetos e arquitetura multicamada.
- A comprovação do uso de teste unitário, padrões de projeto e de programação por contratos será levada em conta na avaliação do trabalho.
- ***Trabalhos copiados resultarão em nota zero para todos os alunos envolvidos.***

#### **F) Entrega do trabalho:**

- Todos os arquivos necessários a execução do sistema, bem como os arquivos-fonte, scripts de banco de dados e os arquivos de teste unitário, deverão ser empacotados em um único arquivo (.zip) e submetidos através do sistema Moodle até a data de entrega.
- Não serão aceitos trabalhos enviados por correio eletrônico.