

# ANÁLISE E DESENHO DE SOFTWARE

## PROJECTO 2021/2022

### PARTE II

O objectivo desta segunda parte do projecto é a implementação dos casos de uso mais relevantes da primeira parte. Deverá pelo menos contemplar as operações de sistema *principais* presentes em **Gerir Leilão** (ainda que tenha fraccionado esse caso de uso em outros mais pequenos, como o **Licitar em Leilão**) e **Criar Leilão** (pois este é indispensável para a aplicação funcionar).

Para tal vai acrescentar o seu código ao *esqueleto* de projecto Maven disponibilizado.

A implementação segue uma estrutura por camadas muito simplificada:

- Na camada de domínio (pacote `business`) deverá incluir as classes que capturam o estado e comportamento dos conceitos analisados nos modelos de domínio e de desenho. Explore os padrões leccionados na disciplina e use aqueles que considerar vantajosos para estruturar ou resolver certas tarefas no projecto. Justifique as suas decisões num pequeno relatório escrito correspondente a esta fase, mas que deve ser anexado ao relatório da 1ª parte.
- Nessa mesma camada de domínio (pacote `business`) poderá colocar classes que sirvam de *handlers*, i.e., representantes dos casos de uso que contêm as tarefas necessárias à execução total ou parcial de cada caso de uso (cf. padrão *GRASP Controller*). O uso de *handlers* é opcional.
- No pacote `client` deverá colocar exemplos de execução. Deve criar uma ou mais classes que demonstrem as funcionalidades que implementou. *Pode* criar uma pequena aplicação interactiva (que, nesse caso, deverá estar acompanhada de instruções de utilização!), mas *deve* sempre criar interações *scriptadas*, que não exijam a intervenção do utilizador do programa durante a execução.
- Por simplicidade, não subdividimos o pacote `business` em subpacotes especializados em partes específicas da aplicação, nem subdividimos o pacote `client` em subpacotes especializados em executar diferentes casos de uso. A complexidade da aplicação nesta fase não o justifica.

A estrutura das classes e as suas relações devem estar de acordo com o modelo de desenho que tenha proposto na primeira parte do projecto.

Deve referir no relatório o conjunto de decisões mais importantes no que toca ao desenvolvimento e teste do seu código.

O código deve vir acompanhado por comentários adequados, incluindo comentários de documentação (a.k.a. *javadocs*) que explicitem o significado das classes e os contratos dos métodos.

Deve igualmente incluir no pacote `src/test/java/business/` uma classe JUnit que efectue testes para classes relevantes relacionadas com os casos de uso que decidiu desenvolver. Para poupar tempo, não precisa de fazer testes exaustivos ao código; portanto, não temos o objectivo de que sejam completos. Usamos a **versão 4** do Junit.

Quando importar o *template* do projecto para o seu Eclipse, faça-o via Maven.

Aproveite o trabalho realizado nas aulas TP sobre Maven, e eventualmente o `pom.xml` aí criado, para permitir que o seu projecto crie um *site* com informação genérica sobre o projecto, os *javadocs*, o resultado dos testes JUnit, e as métricas de cobertura dos testes (via JaCoCo).

Configure a *source* e o *target* da compilação para Java 1.8.

### Entrega

O relatório correspondente a esta parte é a *continuação* do relatório descrito na 1ª parte do enunciado, mas espera-se que seja mais pequeno. Deve portanto manter a formatação e ser incluído no ficheiro pdf com o relatório da 1ª parte.

O projecto Maven desenvolvido no Eclipse deve ser compactado com a sua estrutura de pacotes intacta. Em princípio, bastaria incluir a pasta `src` e o `pom.xml`. Porém, é recomendável incluir também as pastas `.settings` e `target`, e os ficheiros `.classpath` e `.project`, para o caso de surgir algum problema com o *build* Maven no computador do docente e ser preciso comparar com o *build* dos alunos.

O ficheiro zip deve ser entregue no moodle respeitando o mesmo prazo do relatório, ou seja, **até às 23:00 do dia 28 de Maio de 2021**.