

## TPC Prático 2

### Objectivo

O objectivo deste trabalho é modelar (de forma orientada por objectos) e implementar (em Java) uma parte de um sistema informático para suportar uma agência de venda de imobiliário (casas, apartamentos, lojas, etc.).

O sistema terá vários módulos, sendo um deles relacionado com a gestão e visualização de imóveis em carteira e cálculo dos respectivos valores (estimados) de mercado.

Neste TPC iremos modelar e implementar parte deste módulo.

Este trabalho é individual.

### Domínio do problema

Os alunos da Univ. Lusófona foram contratados para desenvolver um sistema para apoio à gestão de uma agência de venda de imobiliário.

A agência tem em carteira um conjunto alargado de imóveis de variados tipos:

- Apartamentos
- Moradias
- Lojas

De seguida é apresentada a informação que a agência regista para cada um dos tipos de imóvel.

<b>Informação</b>
-------------------

Apartamentos:

- Identificador interno (número inteiro)
- Área total (em *metros quadrados*)
- Área útil (em *metros quadrados*)
- Morada (rua, nr da porta, **nr andar, letra**, localidade, país)
- Número de quartos
- Número de WCs
- Valor pretendido (em euros)

Moradias:

- Identificador interno (número inteiro)
- Área total (em *metros quadrados*)
- Área útil (em *metros quadrados*)
- Morada (rua, nr da porta, localidade, país)
- **Número de andares**
- Número de quartos
- Número de WCs
- Valor pretendido (em euros)

Lojas:

- Identificador interno (número inteiro)
- Área total (em *metros quadrados*)
- Área útil (em *metros quadrados*)
- Morada (rua, nr da porta, localidade, país)
- Valor pretendido (em euros)

Para cada um dos imóveis, a agência mantém também a seguinte informação relativa ao seu dono:

- Identificador de cliente (número inteiro)
- Nome
- Apelido
- Contacto telefónico

Pode acontecer que a mesma pessoa seja dona de vários dos imóveis que a agência tem em carteira.

<b>Funcionalidades pretendidas</b>
------------------------------------

As funcionalidades a implementar no sistema são as seguintes:

1. Listagem de todos os imóveis em carteira;
2. Listagem de todos os imóveis em carteira, situados numa determinada localidade;
3. Cálculo do valor (estimado) total dos imóveis em carteira;

As listagens (itens 1 e 2) devem apresentar a informação completa do imóvel, para além do nome completo e contacto telefónico do respectivo dono. Entre cada imóvel, deve ser apresentado um separador (espaço em branco ou caracteres especiais).

Relativamente ao **Cálculo do valor estimado** de cada imóvel, o mesmo deve ser feito considerando as seguintes fórmulas:

- Apartamentos -  $(n^{\circ} \text{ quartos} + n^{\circ} \text{ WCs}) * 300 + \text{área útil} * \text{PPMQ}$
- Moradias -  $\text{área útil} * \text{PPMQ} + n^{\circ} \text{ andares} * 250$
- Lojas -  $\text{área útil} * \text{PPMQ}$

Sendo “PPMQ” o valor de cada metro quadrado, que depende da **Localidade** e deve ser calculado considerando a tabela seguinte:

Localidade	Valor
Lisboa	1000
Porto	900
Faro	900
Outras	750

## Exercícios

### Parte I - Modelação

1) Analise o enunciado e desenhe um diagrama UML com a sua proposta de modelo para suportar o mesmo.

Nota: o seu modelo deve considerar formas de adicionar novos imóveis ah carteira.

2) Justifique os tipos de dados escolhidos para os atributos da(s) classe(s) que fizer(em) parte do seu modelo.

### Parte II - Implementação do modelo

1) Crie um projecto Java com:

- uma classe principal (que inclua a função **main()**)
- as implementações das classes que fazem parte do seu modelo.

### Parte III - Implementação das funcionalidades

1) Implemente as três funcionalidades descritas.

2) Na função **main()** do projecto, crie objectos (e invoque métodos sobre os mesmos) de forma a demonstrar o funcionamento das três funcionalidades descritas.

### Parte IV - Testes unitários automáticos

Pretende-se criar testes automaticos unitarios (JUnit) para o Cálculo do valor (estimado) dos imóveis e da carteira.

Assim, devem ser criados pelo menos 4 testes automáticos:

- Um para o método que faz o cálculo do valor de um Apartamento;
- Um para o método que faz o cálculo do valor de uma Moradia;
- Um para o método que faz o cálculo do valor de uma Loja;
- Um para o método que faz o cálculo do valor de todos os imóveis em carteira;
  - (Neste teste, carteira de imóveis não pode estar vazia.)

Os casos de teste devem ser implementados usando JUnit, tal como foi descrito nas aulas práticas.

<b>Entrega</b>
----------------

A Parte I deverá ser resolvida num documento num formato que permita o desenho do diagrama e a introdução de texto (p.e.: pdf, doc, ppt). Serão também aceites soluções manuscritas e digitalizadas desde que as mesmas sejam claras e não se encontrem rasuradas.

As restantes Partes (II, III e IV) deverão ser resolvidas usando ficheiros *.java*.

Tal como foi explicado nas aulas práticas, os trabalhos devem ser submetidos para um **repositório git privado no bitbucket**. Deve ser usado o mesmo repositório que foi usado para o TPC #1.

Este repositório deverá ser partilhado com o Professor das aulas práticas respectivo. Por efeitos de organização do conteúdo do repositório, os ficheiros devem estar dentro de uma pasta com o nome “**TPC2**”.

**Prazo de entrega:** 12 de Janeiro de 2018 pelas 22h30 (data e hora do último *commit*). Não serão aceites *commits* após esta data e hora.

Aplicam-se as regras da cadeira relativas a cópias.

Recomenda-se que a implementação deste trabalho siga as “**Regras de Estilo**” publicadas no *moodle* de LP2.

**FIM**