Universidade do Minho
Escola de Engenharia

MESTRADO INTREGRADO EM ENGENHARIA INFORMÁTICA

TRABALHO PRÁTICO DE PROGRAMAÇÃO ORIENTADA AOS OBJETOS

# RELATÓRIO PROJETO IMOOBILIÁRIA




TRABALHO REALIZADO PELO GRUPO 11:

- RICARDO CERTO A75315
- GUILHERME GUERREIRO A73860
- MARCELO LIMA A75210

# Índice:

## 1.Introdução :

No âmbito da unidade curricular de Programação Orientada aos Objetos, e com o objetivo de aplicar os conteúdos aprendidos nas aulas teóricas e postos em prática nas aulas prático-laboratoriais, foi-nos proposto desenvolver uma aplicação em BlueJ na linguagem Java que tinha como objetivo simular uma Imobiliária. Pelo que nessa aplicação teríamos de conseguir utiliza-la quer fossemos vendedores ou compradores sendo possível assim aceder a todos os dados de uma Imobiliária.

## 2. Descrição do enunciado do projeto:

O projeto tem como principal objetivo simular todos os processos que podem ser realizados numa agência imobiliária. E para esse efeito deve ser desenvolvida uma aplicação que consiga fazer uma boa gestão de imóveis e tendo em conta ainda que a aplicação irá ter dois utilizadores distintos, os vendedores e os compradores. Cada um deles vão executar opções diferentes na aplicação, pois fazem coisas distintas dentro da imobiliária.

Tanto os utilizadores como os vendedores possuem características comuns como o nome, email, etc… e depois cada um possui várias informações específicas de acordo com o seu tipo de utilizador.

Quanto á parte dos imóveis, podem ser caracterizados como sendo de quatro tipos distintos tendo cada tipo uma informação extra que os distingue dos outros. Para além disso todos os imóveis podem ser selecionados para leilão podendo qualquer comprador registado participar no mesmo. Se um comprador tiver a licitação mais alta no fim do leilão e caso seja superado o preço mínimo aceite do imóvel, o comprador adquire esse imóvel.

## 3. Arquitetura das classes:

**IMOOBILIARIA ->** A classe Imoobiliaria funciona como a classe principal, a qual vai ser responsável por armazenar toda a informação da Imobiliaria. Para isso optamos por defenir como uma das variáveis de instância um **HashMap<Imovel,Vendedor>,** o qual vai mapear todos os Imóveis da Imobiliaria ao respectivo vendedor(para definir o hashCode() do Imóvel utilizamos o id do imóvel, uma vez que é diferente para todos os Imóveis e para o vendedor utilizamos o email). Definimos também um **HashSet<Utilizador>** que vai guardar todos os utilizadores registados na Imobiliária (para o definir o hashCode() utilizamos o email do utilizador). É nesta classe que estão todas as funções que fornecem a interface ao utilizador, desde a função que regista um Imóvel até á função que permite colocar um Imóvel como favorito.

**IMOVEL ->** A classe Imovel é uma classe abstrata. Esta classe tem cinco subclasses, que servem para identificar os diferentes tipos de imóveis com as suas características próprias. Tem como variáveis de instância uma rua, preço pedido, preço mínimo aceite, estado, id e uma **List<Consulta>** que contém todas as consultas ao Imóvel.

**MORADIA ->** É um imóvel de habitação familiar que possui as seguintes características para além daquelas comuns a todos os imóveis, o tipo, a área de implantação , a área total coberta, a área do terreno envolvente , o número de quartos , o número de WC´s , o número da porta , o número de andares que possui e se tem ou não garagem.

**APARTAMENTO ->** É um imóvel de habitação familiar, que possui as seguintes características para além daquelas comuns a todos os imóveis, o tipo, a área total, o número de quartos, o número de WC´s, o número da porta, o número de andares e se possui ou não garagem.

**LOJA ->** É um imóvel de negócios nao habitável. Possui várias características para além daquelas comuns a todos os imóveis, como a área, se possuem WC ou não, o tipo de negócio, o número da porta.

**LOJAHABITAVEL ->** É um imóvel de negócios habitável, possuindo assim todas as caracteristicas de uma loja e tabém de um Apartamento.

**TERRENO ->** Trata-se de um espaço com uma determinada área disponível para construção de um determinado imóvel. O terreno podia ser usado para construção de armazéns ou para construção de habitações e ainda possui as seguintes informações: o diâmetro das canalizações, os kWh máximos suportados pela rede elétrica e se têm acesso á rede de esgotos.

**INTERFACE HABITÁVEL ->** A classe habitável, que se encontra vazia, serve apenas para distinguir os imóveis que são habitáveis dos que não são.(Habitáveis: Moradia, Apartamento, LojaHabitável. Não Habitáveis: Loja, Terreno).

**UTILIZADOR ->** Para os utilizadores criamos uma classe abstrata. Os utilizadores dividem-se em compradores e vendedores, mas ambos possuem características em comum(presentes nesta classe). Essas características são um nome, um email, uma password, uma data de nascimento.

**VENDEDOR ->** Os vendedores possuem um portefólio dos seus imóveis em venda bem como um historial dos seus imóveis vendidos. Optamos por defenir ambos como um **TreeSet<Imovel>**, porque nos pareceu mais adequado uma vez que não existem Imóveis iguais. Utilizamos como metodo de comparação o perço pedido pelo Imóvel.

**COMPRADOR ->** Os compradores possuem uma lista de imóveis favoritos. Optamos por defeni-lo também como um **Tree Set<Imovel>** pela mesma razão que o anterior.

**LICITADOR ->** O licitador é uma subclasse do Comprador e possui como variáveis de instância um limite máximo que está disposto a oferecer, o valor de incrementos a efectuar, o intervalo entre licitações , o id do correspondente comprador e o tempo em milissegundos da sua ultima licitação.

**LEILAO ->** A classe Leilao é responsável por gerir todos os processos de um Leilão. Tem como variáveis de instância um **montante** (que corresponde ao valor da licitação mais alta actual), as **horas** (que corresponde ao tempo do leilão), um **Imovel** (que corresponde ao Imovel que está a ser leiloado), um **List<Licitador>** (lista de todos os licitadores que estão a participar no leilão) e um **Licitador**(que corresponde ao vencedor do leilão). Um vendedor pode colocar um imóvel em leilão, definindo um tempo limite em que vai estar disponível para venda. Durante esse tempo qualquer comprador pode fazer propostas com intuito de comprar o imóvel, desde que antes tenha sido inscrito no leilão e que tenha declarado qual o seu valor máximo a pagar pelo imóvel e o valor constante de quanto é queria acrescentar á sua proposta de cada vez e em período de tempo faz uma nova licitação.

**CONSULTA ->** A classe Consulta tem como variáveis de instância um **LocalDateTime** que corresponde á data da consulta e um **email** do utilizador que efectuou a consulta(caso esteja registado). Sendo assim, esta classe serve para criar um registo das consultas que um determinado utilizador fez a um determinado Imóvel.

**IMOOBILIARIA APP ->** É a classe responsável pela interface com o utilizador, efectuando todos os inputs e outputs da aplicação.

## 4. Desenvolvimento da Aplicação:

Esta aplicação foi desenvolvida com o intuito de dar a possibilidade a vários utilizadores de navegar livremente por todas as funcionalidades oferecidas por uma Imobiliária. Deste modo, é necessário fornecer uma interface que seja intuitiva e fácil de utilizar. Por esta razão, desenvolvemos uma interface de utilizador que possui apenas 3 tipos de menus que vao passar a ser expostos de seguida.

## Menu Principal

```
    |========== IMOBILIARIA ===========|
1 - Registar Conta
2 - Iniciar Sessão
3 - Lista de Imóveis de um dado tipo
4 - Lista de Imóveis Habitáveis
5 - Mapeamento Imóvel-Vendedor
0 - Sair
Opção:
```

É este o Menu que irá aparecer no momento em que a aplicação é executada. Como é possível observar, temos ao nosso dipor várias funcionalidades, como registar uma nova conta, iniciar sessão, consultar a lista de todos os Imóveis de um certo tipo,

até um certo preço, registados na Imobiliária, consultar a lista de todos os Imóveis habitáveis, até um certo preço, registados na Imobiliária e por fim obter um mapemento entre cada Imóvel e o respectivo vendedor.

## Menu dos Vendedores

```
    |========== IMOBILIARIA ===========|
       Sessão iniciada - 'Vendedor1'
1 - Registar Imóvel
2 - Últimas 10 Consultas aos seus Imóveis em Venda
3 - Alterar estado de Imóvel
4 - Os seus Imóveis mais consultados
5 - Lista de Imóveis de um dado tipo
6 - Lista de Imóveis Habitáveis
7 - Mapeamento Imóvel-Vendedor
8 - Iniciar Leilão
9 - Terminar Sessão
0 - Sair
Opção:
```

Será este o Menu que aparecerá caso um utilizador inicie sessão como vendedor. Novamente, como é possivel observar existem várias funcionalidades ao dispor do vendedor, bastantes mais do que as de um utilizador não registado. Como vendedor, é possível registar um Imóvel de vários tipos, no entanto tem de ser um dos tipos abrangidos pela Imobiliária(Moradia, Loja, Terreno, LojaHabitavel e Apartamento). Um vendedor pode visualizar as 10 últimas consultas aos seus Imóveis em venda, sendo-lhe mostrado a data dessa consulta assim como o email de quem a consultou(caso possua). Pode ainda alterar o estado de um dos seus Imóveis(Estados: Venda, Vendido, Reservado). Pode visualizar os seus Imóveis que possuem mais de n consultas e para além de poder fazer tudo que um utilizador nao registado consegue, também é capaz de iniciar um leilão com um dos seus Imóveis, defenindo um periodo para o mesmo.

## Menu dos Compradores

```
    |========== IMOBILIARIA ===========|
       Sessão iniciada - 'Comprador1'
1 - Marcar Imóvel como favorito
2 - Lista dos seus Imóveis favoritos
3 - Lista de Imóveis de um dado tipo
4 - Lista de Imóveis Habitáveis
5 - Mapeamento Imóvel-Vendedor
6 - Terminar Sessão
0 - Sair
Opção:
```

Será este o Menu que aparecerá caso um utilizador inicie sessão como comprador. Novamente, como é possivel observar existem várias funcionalidades ao dispor do comprador, bastantes mais do que as de um utilizador não registado. Como comprador, é possível marcar um determinado Imóvel como favorito, visualizar a lista dos seus Imóveis favoritos e fazer tudo aquilo que um utilizador normal pode fazer.

## 5. Inclusão de novos tipos de imóveis na aplicação:

Um determinado utilizador não consegue criar um tipo de imóvel para além daqueles que já estão pre-defenidos na aplicação. Desta forma, se isto acontecer, para incluir novos tipos de imóveis na aplicação, bastava-nos criar uma nova classe que corresponde a um novo tipo de imóvel, com todas as informações necessárias e com os seus métodos de instância, de seguida bastava coloca-la como subclasse da classe Imovel e adicionar essa opção aos menus, mais concretamente ao menu dos vendedores.

## 6. Conclusão:

Com a realização deste projeto conseguimos aplicar grande parte dos conhecimentos que obtivemos nas aulas de POO e pensamos que o resultado está bastante perto do esperado. Foi bastante útil na consolidação dos conhecimentos e funcionalidades que a liguagem de programação Java nos consegue oferecer. Desta forma, de certeza que o resultado e empenho que demonstramos neste trabalho, nos vai vir a ser bastante útil num futuro profissional.