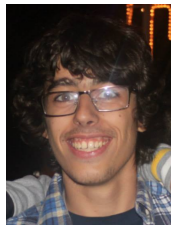


Relatório do Projeto ”Imoobiliária”

Programação Orientada aos Objectos

Autores:



Bruno Cancelinha
A75428



José Bastos
A74696



Marcelo Miranda
A74817

TRABALHO REALIZADO PELO GRUPO 10

MESTRADO INTEGRADO EM ENGENHARIA
INFORMÁTICA - 2º ANO - 2º SEMESTRE



Universidade do Minho

20 de Maio de 2016

Conteúdo

| | | |
|----------|---|-----------|
| 1 | Introdução | 1 |
| 2 | Arquitetura das classes | 2 |
| 2.1 | ImoobiliáriaApp | 2 |
| 2.2 | Imoobiliária | 2 |
| 2.3 | Utilizador | 3 |
| 2.3.1 | Comprador | 3 |
| 2.3.2 | Vendedor | 3 |
| 2.4 | Imóvel | 3 |
| 2.4.1 | Moradia | 4 |
| 2.4.2 | Apartamento | 4 |
| 2.4.3 | Loja | 4 |
| 2.4.4 | LojaHabitável | 4 |
| 2.4.5 | Terreno | 5 |
| 2.5 | Menu | 5 |
| 2.6 | Habitável | 5 |
| 2.7 | Consulta | 5 |
| 2.8 | Leilão | 5 |
| 2.8.1 | Licitador | 6 |
| 3 | Descrição da aplicação | 7 |
| 4 | Possíveis novos tipos de imóveis | 11 |
| 5 | Conclusão | 12 |

Resumo

Neste relatório é apresentada a descrição da realização da aplicação solicitada do projeto de Programação Orientada aos Objectos, que tem como objetivo a criação de um programa de gestão de imóveis.

1. *Introdução*

Este projeto foi realizado no âmbito da disciplina de Programação Orientada aos Objectos e tem como objetivo a criação de uma aplicação de gestão de imóveis, de forma a simular uma agência imobiliária, abrangendo desde a criação de um imóvel sob forma de um anúncio até ao registo da sua venda. Nesta aplicação existem dois tipos de atores: os compradores e os vendedores. Os compradores podem pesquisar imóveis de acordo com as suas características preferenciais e ainda marcar certos imóveis como favoritos. Já os vendedores conseguem controlar os anúncios dos imóveis, assim como alterar o seu estado. Ainda conseguem aceder a informação sobre estatísticas dos imóveis e anúncios. Este projeto consiste na programação da aplicação na linguagem Java e nas várias tarefas que nos foram propostas usando a matéria que nos foi lecionada na cadeira durante o semestre.

2. *Arquitetura das classes*

Neste capítulo expô-mos todas as classes, atributos e o funcionamento de cada uma ainda como algumas decisões que tivemos que tomar.

2.1 ImobiliáriaApp

É a classe responsável pela interface com o utilizador, e gravar e ler o estado da aplicação.

Atributos:

- Imobiliaria empresa - A imobiliária que corre na aplicação.
- Menu menuPrincipal - Menu principal
- Menu menuComprador - Menu para compradores
- Menu menuVendedor - Menu para vendedores
- Menu menuRegistar - Menu para registar utilizador
- Menu menuImovel - Menu para registar imóvel

2.2 Imobiliária

Na classe Imobiliária estão desenvolvidas todas as funções necessárias para o funcionamento da imobiliária, estando ligado a todas as outras classes.

Atributos:

- Utilizador utilizador - Utilizador que está com sessão iniciada
- Map <String, Utilizador>utilizadores - Lista dos utilizadores.
- Map <String, Imovel>anuncios- Lista dos imóveis disponíveis na Imobiliária.

2.3 Utilizador

Classe abstrata que contém a informação do email, nome, password, morada e data de nascimento.

Atributos:

- String email, nome, password, morada e LocalDate dataNascimento - dados necessários do utilizador

2.3.1 Comprador

Classe que para além dos atributos herdados da classe Utilizador é acrescentada uma lista de imóveis favoritos.

Atributos:

- Set <Imovel>favoritos - Imóveis favoritos de um comprador.

2.3.2 Vendedor

Classe que para além dos atributos herdados da classe Utilizador, os vendedores têm acesso a um portfólio de imóveis bem como ao acesso do historial das 10 últimas vendas efetuadas.

Atributos:

- Map <String, Imovel>imoveisEmVenda - portfólio dos imóveis disponíveis para venda
- Map <String, Imovel>imoveisVendidos - Historial dos imóveis vendidos pelo vendedor
- Leilao leilao - Leilão que decorre

2.4 Imóvel

A classe imóvel é uma classe abstrata a que todos os imóveis podem aceder a esta informação, servindo para diferir dos diferentes tipos de imóveis com as suas características próprias. Nos métodos desta classe estão incluídos os métodos mais usados como construtores, get's, set's, equals, toString e clone.

Atributos:

- String id, rua, estado - Dados necessários para o imóvel
- int precoMinimo, precoPedido - Preço minimo aceite pelo vendedor e preço pedido do imóvel
- List <Consulta>consultas - Lista de consultas dos imóveis

2.4.1 Moradia

Classe que corresponde um Imóvel do tipo Moradia, que para além dos atributos comuns do Imóvel ainda é adicionado a informação sobre o tipo de moradia, a área de implantação, área total coberta, área de terreno, número de quartos e WCs e número da porta.

Atributos:

- String tipo, double areaImplantacao, areaCoberta, areaTerreno, int quartos, wc, porta - dados necessários para o imóvel Moradia

2.4.2 Apartamento

Classe que corresponde um Imóvel do tipo Apartamento, que para além dos atributos comuns do Imóvel ainda é adicionado a informação sobre o tipo de apartamento, área total, número de quartos e WCs e número da porta e andar e se possui garagem.

Atributos:

- String tipo, double area, int quartos, wc, andar, porta, boolean garagem - dados necessários para o imóvel Apartamento

2.4.3 Loja

Classe que corresponde um Imóvel do tipo Loja, que para além dos atributos comuns do Imóvel ainda é adicionado a informação sobre o tipo de negócio, área total, se possui WC e número da porta.

Atributos:

- String tipoNegocio, double area, int porta, boolean wc - dados necessários para o imóvel Loja

2.4.4 LojaHabitável

Igual à classe Loja apenas adicionada a informação que corresponde à classe Apartamento.

Atributos:

- String tipoApartamento, double areaApartamento, int quartos, wcApartamento, andar, boolean garagem - dados necessários para o imóvel LojaHabitável

2.4.5 Terreno

Classe que corresponde um Imóvel do tipo Terreno, que para além dos atributos comuns do Imóvel ainda é adicionado a informação sobre o tipo de terreno, a área disponível para construção, o diâmetro das canalizações, os kWh máximos suportados pela rede elétrica e se possui saneamento.

Atributos:

- String tipo, double area, diametroCanalizacao, potenciaSuportada, boolean acessoEsgotos - dados necessários para o imóvel Terreno

2.5 Menu

Classe que implementa todos os menus e funcionamento dos mesmos, estando relacionado com a classe ImoobiliáriaApp.

Atributos:

- int op - Opção escolhida pelo utilizador
- List <String>opcoes - Lista de opções do menu

2.6 Habitável

A classe habitável serve apenas para distinguir os imóveis que são habitáveis dos que não são.

2.7 Consulta

Classe que serve para criar um registo de consultas, sendo armazenada informação sobre o utilizador e a hora da consulta.

Atributos:

- LocalDateTime data - data da consulta
- String email - email do utilizador que fez a consulta

2.8 Leilão

Classe que é responsável por controlar todo o processo do leilão.

Atributos:

- Imovel im - imóvel em questão no leilão

- int horas - número de horas que o leilão demora
- boolean começou - determina se o leilão já começou
- List <Licitador>licitadores - Lista de licitadores

2.8.1 Licitador

O licitador é o responsável por fazer as licitações nos leilões.

Atributos:

- String idComprador - ID do comprador
- montante, limite, incrementos, minutos - montante, montante máximo possível, incremento a fazer em cada licitação, tempo entre cada licitação
- long ultimaLicitacao - o tempo da ultima licitação

3. *Descrição da aplicação*

Desenvolvemos uma aplicação com uma interface amigável para o utilizador com comandos básicos, para que este pudesse tirar o maior proveito deste. Os menus funcionam à base de opções por números, facilitando assim, a nosso ver, o funcionamento do programa. Quando um utilizador corre inicialmente o programa depara-se com este menu:

```
*** Menu ***  
1 - Iniciar Sessão  
2 - Registrar utilizador  
3 - Procurar imóveis por tipo  
4 - Procurar imóveis habitáveis  
5 - Mapear imóveis  
0 - Sair  
Opção: _____
```

Aqui, o utilizador pode iniciar sessão (opção 1), ou caso não tenha feito, registar-se (opção 2), procurar imóveis por tipo (opção 3) ou habitáveis (opção 4), mapear imóveis (opção 5) ou sair da aplicação (opção 0). Se o utilizador decidir efetuar o registo será apresentado o seguinte menu, onde pode escolher se pretende registar-se como Comprador ou Vendedor:

```
*** Menu ***  
1 - Registrar comprador  
2 - Registrar vendedor  
0 - Sair  
Opção:
```

Em ambos os casos, é pedido ao utilizador que dê a informação sobre o seu nome, email, password, morada e data de nascimento.

```
Email: email  
Nome: nome  
Password: password  
Morada: morada  
Data de nascimento (dd-MM-yyyy): 01-01-2000
```

Caso o utilizador que está a ser registado já exista aparece a seguinte mensagem:

Utilizador exemplo já existe

Caso a password não corresponde ao email introduzido aparece a seguinte mensagem:

Password não corresponde

Depois de se ter a sessão iniciada como comprador aparece o menu:

***** Menu *****
1 – Procurar imóveis por tipo
2 – Procurar imóveis habitáveis
3 – Mapear imóveis
4 – Adicionar imóvel aos favoritos
5 – Obter lista de favoritos
6 – Fechar sessão
Opção:

Das opções apresentadas, as opções 1,2 e 3 podíamos executá-las mesmo sem estar com a sessão iniciada. No entanto, se estivermos com ela iniciada pode-se agora adicionar um certo imóvel aos favoritos (opção 4), obter a lista dos imóveis favoritos (opção 5) ou simplesmente fechar a sessão (opção 6).

Procurando por imóveis pelo seu tipo, pede-se o tipo de imóvel e o seu preço máximo:

Opção: 1
Tipo de imóvel: Apartamento
Preço máximo: 5000

Procurando por imóveis habitáveis apenas se pede o seu preço máximo:

Opção: 2
Preço máximo: 10000

Quando se pretende adicionar um imóvel aos favoritos o programa pede o ID do imóvel.

Depois de se ter a sessão iniciada como vendedor aparece o menu:

```
*** Menu ***
1 - Procurar imóveis por tipo
2 - Procurar imóveis habitáveis
3 - Mapear Imóveis
4 - Registrar imóvel
5 - Obter últimas consultas
6 - Obter imóveis mais consultados
7 - Mudar estado de imóvel
8 - Iniciar leilão
9 - Adicionar comprador ao leilão
10 - Encerrar leilão
11 - Fechar sessão
Opção:
```

Sendo que as 3 primeiras opções já foram identificadas, surgem agora novas opções como:

- Registrar um imóvel no sistema (opção 4)

```
*** Menu ***
1 - Moradia
2 - Apartamento
3 - Loja
4 - Loja habitável
5 - Terreno
0 - Sair
Opção: |
```

Sendo que fica aqui apresentado um exemplo de um registo de um imóvel:

```
Opção: 1
Rua: Rua Nova Santa Cruz
Tipo: isolada
Porta: 32
Número de quartos: 9
Número de casas de banho: 140
Área de implantação: 10000
Área coberta: 5000
Área do terreno: 200000
Preço mínimo: 500000
Preço pedido: 750000
```

- Obter últimas consultas (opção 5)
- Obter imóveis mais consultados (opção 6)

- Mudar estado de imóvel (opção 7)
- Iniciar leilão (opção 8)
- Adicionar comprador ao leilão (opção 9)
- Encerrar leilão (opção 10)

4. *Possíveis novos tipos de imóveis*

Depois de uma breve análise, consideramos que seria possível incluir novos tipos de imóveis no programa, bastando para isso criar uma nova classe que herdava da classe abstrata Imóvel todas as informações essenciais, visto que o nosso programa foi desenvolvido para que qualquer eventual alteração fosse fácil de aplicar.

5. *Conclusão*

Neste trabalho acabamos por finalizar a gestão de imóveis Imobiliária aplicando o conteúdo aprendido na aula, ou seja, através de uma série de requisitos que nos foram propostos, conseguimos obter o resultado pretendido sendo possível visualizar o resultado da aplicação através da interface do programa.