

Universidade do Minho

Trabalho Prático Programação Orientada aos Objetos

(2ºano, Licenciatura em Ciências da Computação)

ImOObiliária

Relatório de Desenvolvimento
Grupo 3



Tiago Loureiro (A71191) – Luís Silva (A72113) – Leonel Gonçalves (A72305)

Descrição do Enunciado.

Este trabalho tem como objetivo o desenvolvimento de uma aplicação de gestão de uma imobiliária, uma espécie de rede social com dois tipos de utilizadores (Comprador e Vendedor), onde se pode criar, alterar e visualizar anúncios de compra e venda de imóveis.

A aplicação permite o registo de um utilizador (Vendedor ou Comprador), assim como iniciar sessão.

O utilizador comprador poderá pesquisar anúncios de imóveis para compra, e o utilizador vendedor é que terá a função de criar/alterar os anúncios.

A aplicação também deverá conter outras funcionalidades como consultar um determinado imóvel, tanto como uma pesquisa por certas características, como por exemplo pesquisar imóveis por tipo.

Arquitetura das classes.

Classe Imovel - É uma classe abstrata que contém as classes dos diferentes imóveis e as variáveis comuns a todos (rua, preço pedido, preço aceite e o numero de consultas).

Classes Apartamento, Loja, Terreno, Moradia, LojaHabitavel – São classes-filhas da classe abstrata Imovel que contém as variáveis específicas de cada uma mais as variáveis comuns que estão na classe Imovel. Todas estas classes servem para criar os vários tipos de imóveis propostos.

Classe Actores – É uma classe abstrata que contém as classes dos diferentes utilizadores e as variáveis comuns a todos.

Classes Vendedor e Comprador – São classes-filhas da classe Actores que contém as variáveis específicas de cada uma mais as variáveis da classe Actores. Estas classes servem para criar os utilizadores propostos com as devidas permissões e restrições.

Classe Imobiliaria – É a classe onde estão implementados os métodos das diversas opções de funcionamento da aplicação.

Nesta classe está implementado o método que apresenta os diferentes menus da aplicação, assim como os métodos que permitem registar um imóvel, registar um utilizador, iniciar sessão e manipular/aceder à informação.

Na classe Imobiliaria optamos por ter uma String logado que guarda o email do utilizador que está com sessão iniciada no momento, optamos também pelo TreeMap tanto para guardar os imóveis como os actores.

Para o TreeMap dos imóveis a chave utilizada foi um Integer que representa o id do imóvel e o valor é um objecto do tipo Imovel.

Para o TreeMap dos actores a chave é uma String que guarda o email do utilizador e o valor é um objecto do tipo Actores.

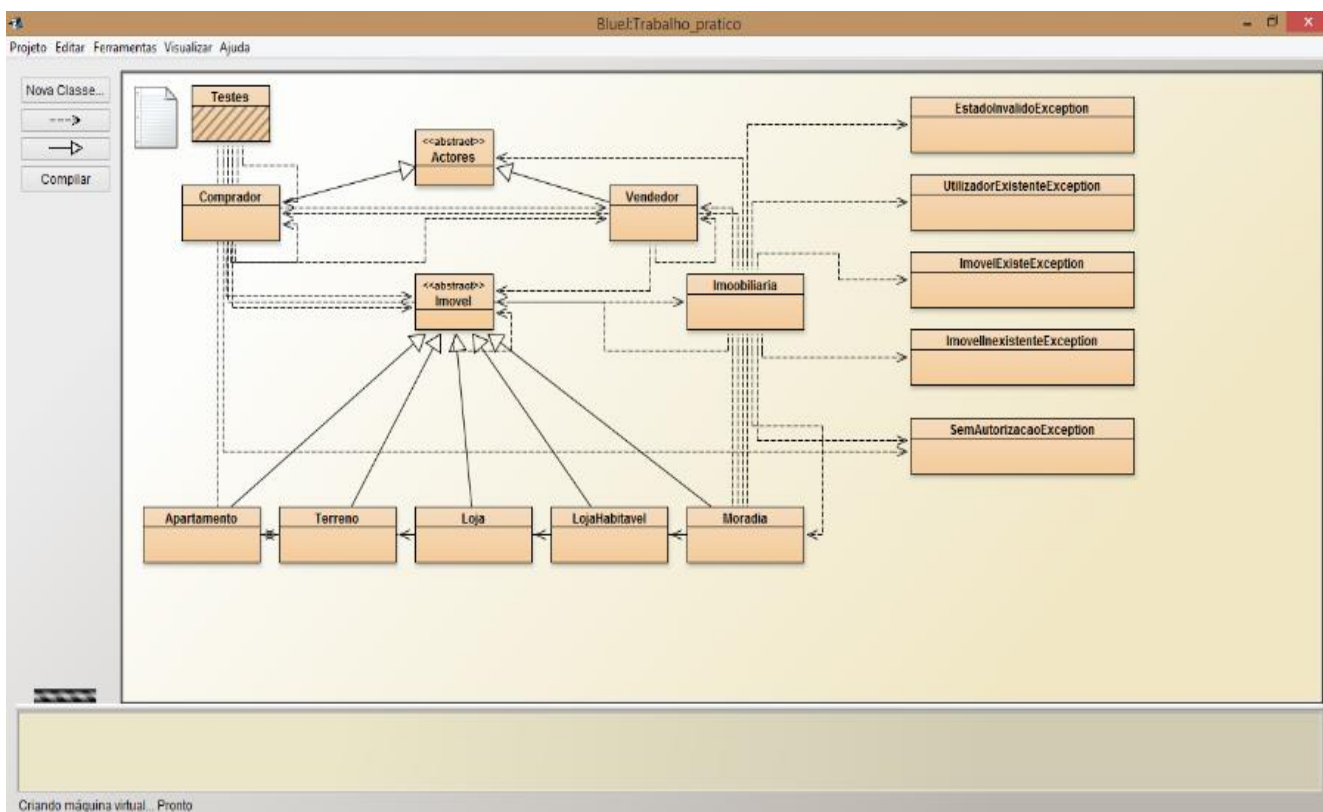
A escolha desta chave e deste TreeMap foi para nos simplificar a vida porque assim tínhamos um tipo fácil de manipular a representar cada imóvel/actor.

Explicação da aplicação desenvolvida.

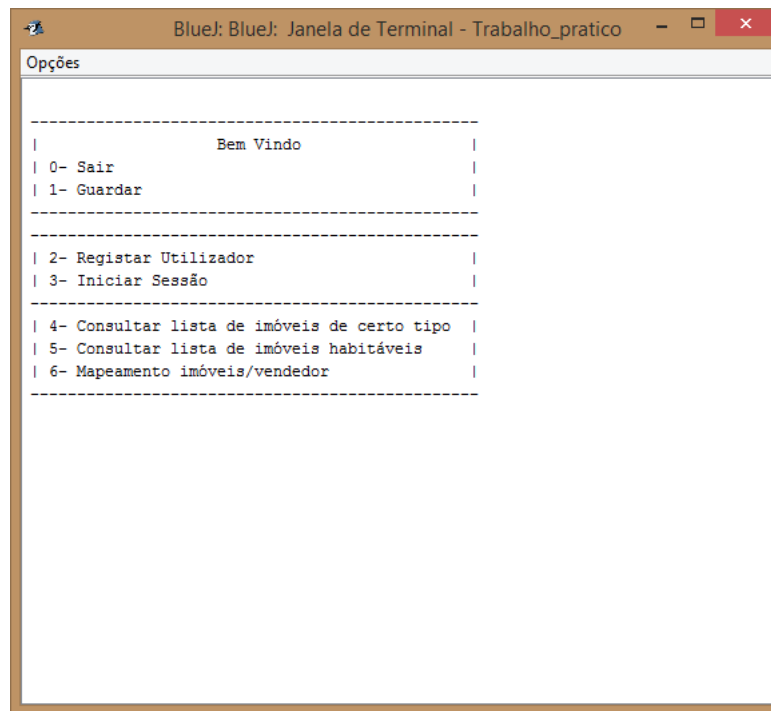
Todos os utilizadores são capazes de consultar a lista de um tipo específico de imóveis até um certo preço, consultar uma lista com todos os imóveis habitáveis e obter um mapeamento com os imóveis e respetivos vendedores.

Ao Vendedor é permitido colocar/alterar um imóvel à venda, visualizar uma lista com as datas das últimas consultas aos imóveis que tem à venda e alterar o estado de um imóvel (por exemplo, passar da lista de imóveis à venda para a lista de vendidos).

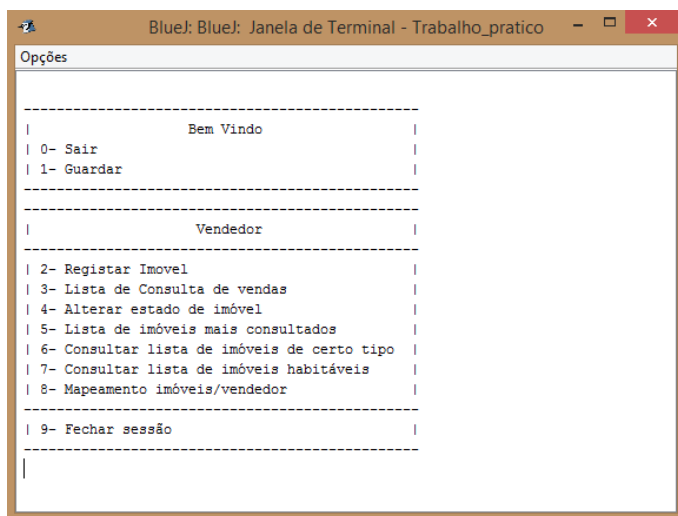
Ao Comprador é permitido marcar imóveis como favoritos.



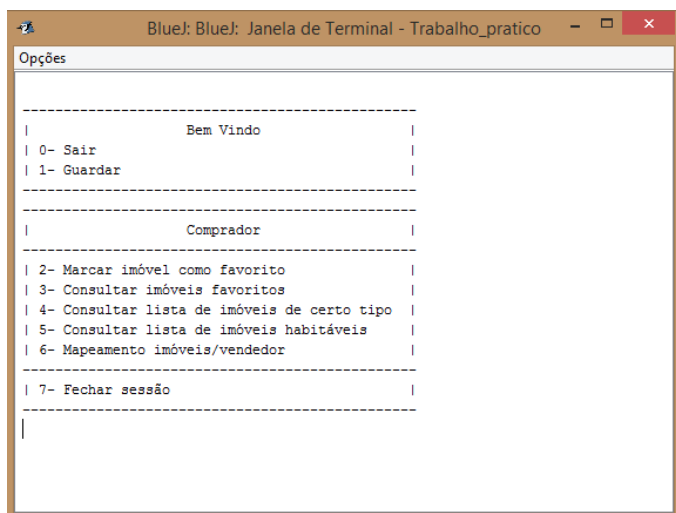
(Esta imagem mostra a maneira de como as classes estão ligadas.)



(Menu inicial.)



(Menu exclusivo dos vendedores.)



(Menu exclusivo dos compradores.)

Seria possível incluir novos tipos de imóveis na aplicação?

Para acrescentar novas classes de imóveis seria um processo bastante simples. Bastava criar a class e esta herdar as variáveis da class Imovel, isto em termos do que é a aplicação sem a interação utilizador/máquina, porque se fosse para tornar a aplicação completamente funcional seria um processo cansativo de acrescentar linhas de código à class Imobiliaria.