





ImOObiliária

Programação Orientada aos Objetos

		
Ana Isabel Castro a55522	Joana Miguel a57127	Lúcia Abreu a71634

Grupo 23

Braga, 20 de Maio de 2016

Data de conclusão do projeto	20 de Maio de 2016
Unidade Curricular	Programação Orientada aos Objetos
Coordenador da Unidade Curricular	José Francisco Creissac Freitas Campos
Equipa Docente	José Francisco Creissac Freitas Campos
	António Luís Pinto Ferreira Sousa
	João Alexandre Baptista Vieira Saraiva
	Rui Miguel Silva Couto
Ano Letivo	2015/2016

1. Conteúdo

2. Introdução	3
3. Considerações Importantes	4
4. Arquitetura de Classes	5
Classe Utilizador <<abstract>>	5
Classe Vendedor	6
Classe Comprador	6
Classe Imovel <<abstract>>	7
Interface Habitável	7
ComparadorPreco	8
Consulta	8
Classe Terreno	8
Classe Apartamento	9
Classe Loja	10
Classe Loja com Apartamento	10
Classe Moradia	11
Classe Imobiliária	12
Interface Imobiliável	12
Classe Leilao	13
Exceções	14
5. Interface de Utilizador	15
Página Inicial	16
Página Inicial Comprador	19
Página Inicial Vendedor	21
Menús da Página Inicial Vendedor	23
Leilão	23
6. Discussão – Novos tipos de imóveis na aplicação	25

2. Introdução

ImOObiliária é um projeto realizado no âmbito da unidade curricular de Programação Orientada aos Objetos do 2ºano do Mestrado Integrado em Engenharia Informática da Universidade do Minho.

O objetivo desta aplicação é fazer a gestão de imóveis, dando suporte a todo o desenvolvimento de um imóvel numa imobiliária. Possui um ambiente de recolha de informação que permite que os seus utilizadores – Vendedores e Compradores – consigam registar e vender imóveis, consultar imóveis segundo as suas características, marcá-los como favoritos, entre outros.

Cada utilizador possui a seguinte informação pessoal: email, nome, password, morada e data de nascimento. O comprador poderá definir a sua lista de imóveis favoritos e o vendedor poderá possuir um portfolio de imóveis em venda e um histórico de imóveis vendidos.

Na ImOObiliária existem quatro tipos de imóveis: Moradias, Apartamentos, Lojas e Terrenos.

Existe a possibilidade de um vendedor colocar um seu imóvel em Leilão, que consiste num período de tempo durante o qual o imóvel é leilado ao público. Os compradores podem fazer licitações até ao fecho do leilão. No final é indicado qual o utilizador que compra o leilão, assim como o valor da sua venda, isto se, atingiu o preço mínimo.

3. Considerações Importantes

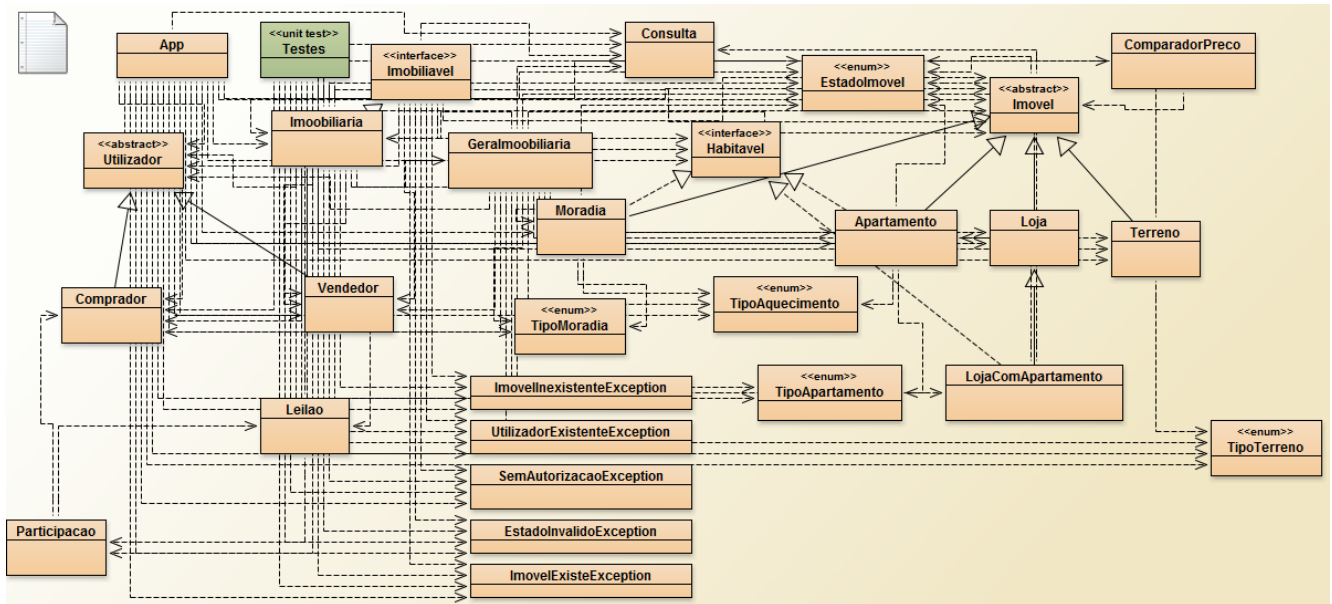
- Definiu-se que cada imóvel só deverá estar associado a um vendedor, ao qual, cabe “promover” a sua venda (logo, dois imóveis para venda nunca poderão estar associados ao mesmo vendedor).

- Ficou definido que cada utilizador tem associado um e-mail único, com o qual faz log in no sistema. Logo o e-mail é o elemento que permite identificar o utilizador.

- No sistema optou-se por definir que um utilizador ou é comprador ou vendedor, nunca poderá ser os dois em simultâneo

- Também o imóvel tem o seu elemento identificador único, denominado código de imóvel.

4. Arquitetura de Classes

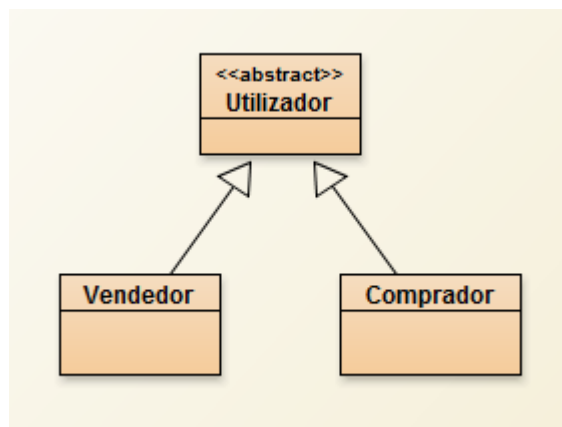


Classe Utilizador <<abstract>>

Variáveis de Instância

```
private String email;
private String nome;
private String password;
private String morada;
private LocalDate dataNascimento;
```

A classe Utilizador possui as variáveis de instância email, nome, password, morada e data de nascimento, que são as classes que o Vendedor e Comprador irão herdar.



Nas subclasses Vendedor e Comprador decidiu-se utilizar a estrutura de dados em que cada elemento é uma <String> e não um objeto <Imovel>. Se a escolha tivesse sido <Imovel>, cada vez que um imóvel fosse modificado na classe Imovel, ter-se-ia de atualizar a lista de imóveis das classes Vendedores e Compradores.

Assim, com <String>, estas classes apenas possuem o código do imóvel, e cada vez que precise de aceder a um imóvel, pode-se obtê-lo atualizado da classe Imovel. A <String> representa um código único e identificativo do imóvel.

Classe Vendedor

- **Imóveis Para Venda (Portfolio)**

A estrutura de dados escolhida para o portfolio de imóveis para venda é ArrayList<String> com o código de cada imóvel.

Decidiu-se utilizar <ArrayList> pois na opinião do grupo, o vendedor tem interesse em consultar os seus imóveis para venda do mais antigo para o mais recente, para “terminar” ou “vender” primeiro os negócios começados há mais tempo. Nestes imóveis para venda estão incluídos os imóveis reservados. Assim sendo, os imóveis para venda possuem dois estados: “RESERVADO” e “PARA VENDA”. Quando um imóvel é adicionado, é garantido que não há repetidos.

- **Imóveis Vendidos (Histórico)**

A estrutura de dados escolhida para o histórico de imóveis vendidos é ArrayList<String> com o código de cada imóvel.

Decidiu-se utilizar <ArrayList> pois num histórico é importante que haja uma noção de ordem, neste caso a ordem pela qual foram vendidos. Quando um imóvel é adicionado, é garantido que não há repetidos.

Classe Comprador

- **Imóveis Favoritos**

A estrutura de dados escolhida para o portfolio de imóveis para venda é ArrayList<String> com o código de cada imóvel.

Decidiu-se utilizar <ArrayList> pois permite guardar por ordem de adição à lista de favoritos. Assumiu-se que o utilizador pretende ver a sua lista de preferidos pela ordem que adicionou os imóveis. Quando um imóvel é adicionado, é garantido que não há repetidos.

Notas:

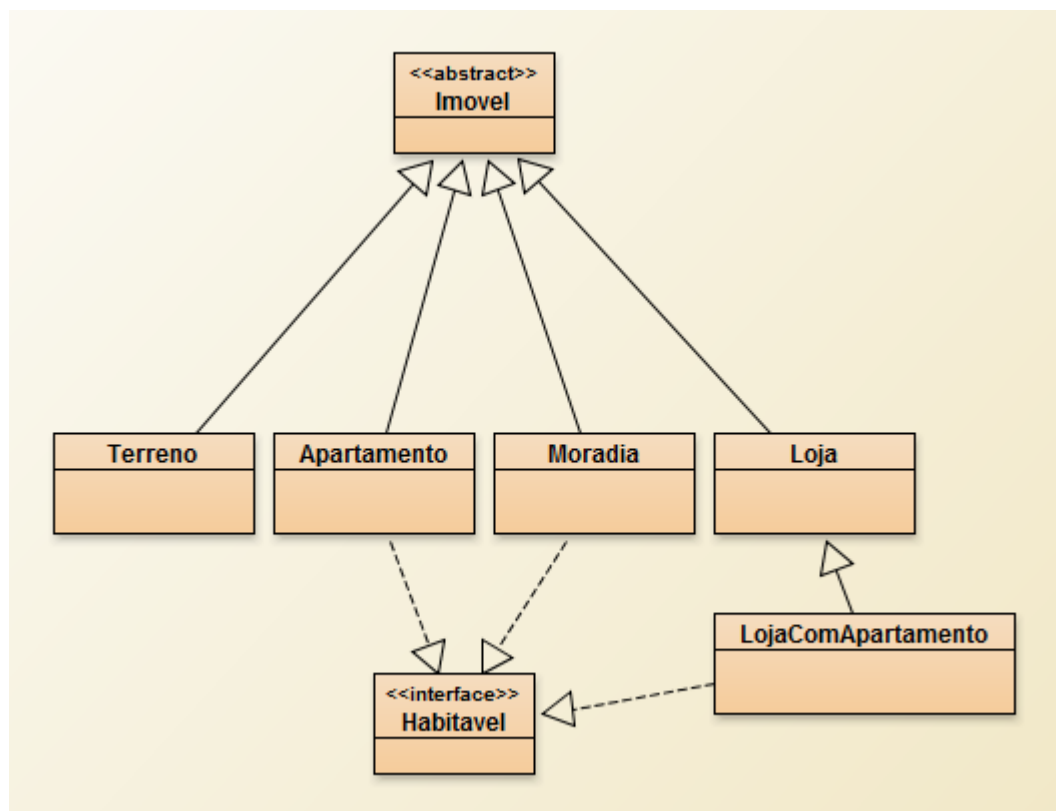
- Ficou definido que cada utilizador tem associado um email único, com o qual faz login no sistema. Logo, o email é o elemento que permite identificar o utilizador.
- No sistema, optou-se por definir que um utilizador ou é comprador ou vendedor, nunca poderá ser os dois em simultâneo.

Classe Imovel <<abstract>>

Variáveis de Instância

```
private String codigo;  
private String rua;  
private int precoPedido;  
private int precoMin;  
private EstadoImovel estado;  
private List<Consulta> consultas;  
private String vendedor;
```

A classe abstrata Imovel possui as variáveis de instância código (do imóvel), rua, precoPedido, precoMin (que nunca será apresentado ao comprador), estado (VENDA, RESERVADO, VENDIDO), consultas (as consultas feitas a cada imóvel) e vendedor. As classes Terreno, Apartamento, Moradia e Loja irão herdar estas variáveis de instância.



Interface Habitável

"A picture is worth a thousand words. An interface is worth a thousand pictures."

Ben Shneiderman

A interface permite reunir similaridades comportamentais, entre classes não relacionadas hierarquicamente. Pelo que pode dizer-se que uma interface é um contrato que quando assumido por uma classe deve ser implementado.

As classes Moradia, Apartamento e Loja com apartamento são Habitáveis, a interface Habitável é um marcador que permite marcar estas classes como habitáveis. As classes que queiram pertencer a esta interface são obrigadas a implementar o método `getTipologia()`.

ComparadorPreco

O comparador ComparadorPreco fornece um método de comparação dos diferentes imóveis, pelo seu preço, e é essencial para os métodos que consultam imóveis até um determinado preço.

Permite assim organizar os imóveis segundo essa informação.

O comparador natural para os imóveis é o comparador por código de imóvel.

Consulta

Variáveis de Instância

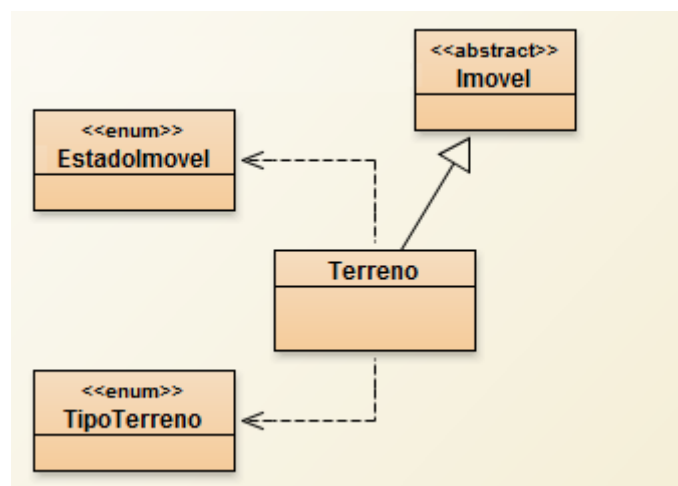
```
private String email;  
private LocalDateTime dataConsulta;
```

Sempre que um imóvel ou o respetivo código do imóvel é devolvido num método próprio do comprador (ou utilizador não registado mas possível comprador), e como essa informação interessa ao vendedor (saber os imóveis mais apelativos aos compradores), é considerado uma consulta.

Classe Terreno

Variáveis de Instância

```
private int areaDisponivel;  
private TipoTerreno tipo;  
private int diametroCanalizacoes;  
private boolean redeInstalada;  
private int potenciaMax;  
private boolean redeEsgotos;
```



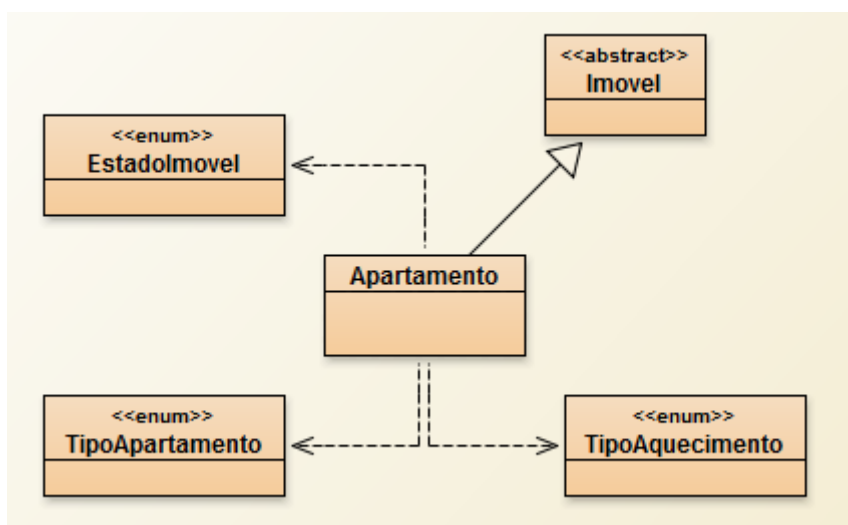
A Classe Terreno é uma classe que herda da Classe Imovel, e possui as variáveis de instância que herdou desta, e ainda as seguintes: areaDisponivel, tipo, diametroCanalizacoes, redeInstalada, potenciaMax e redeEsgotos.

A variável de instância tipo definiu-se como tipo Enum (apenas sendo possível definir terrenos para HABITACAO ou ARMAZEM).

Classe Apartamento

Variáveis de Instância

```
private TipoApartamento tipo;  
private int areaTotal;  
private int numQuartos;  
private int numWCs;  
private int numPorta;  
private int numAndar;  
private int anoConstrucao;  
private TipoAquecimento tipoAquecimento;  
private boolean garagem;  
private int areaGaragem;
```



A Classe Apartamento é uma classe que herda da Classe Imovel, e possui as variáveis de instância que herdou desta, e ainda as seguintes: tipo, areaTotal, numQuartos, numWCs, numPorta, numAndar, anoConstrucao, tipoAquecimento, garagem e areaGaragem.

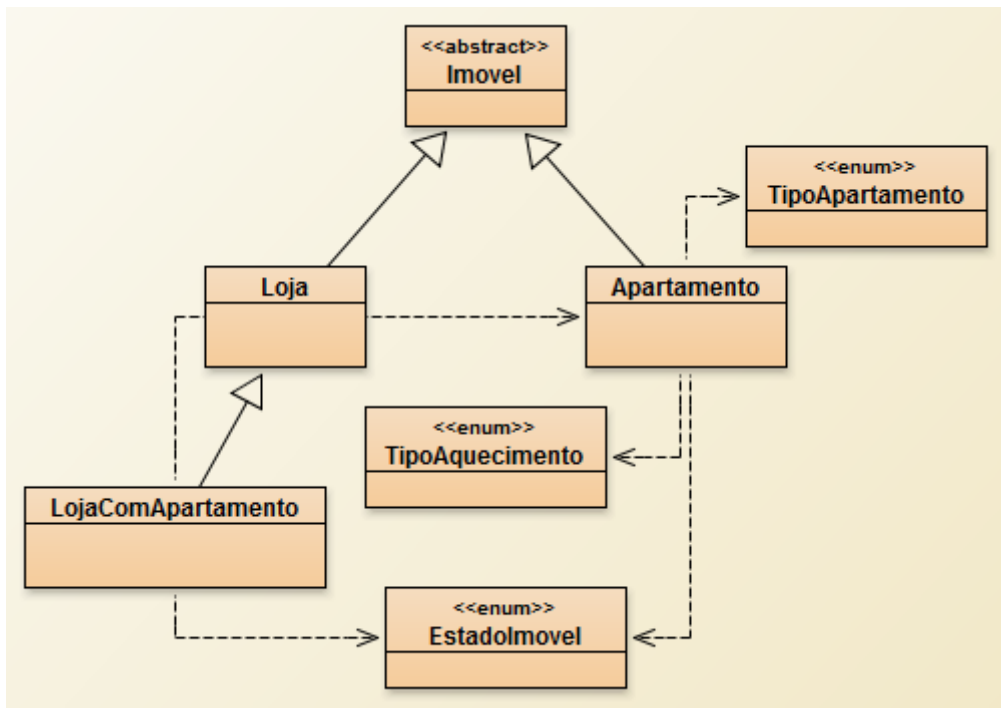
A variável de instância tipo definiu-se como tipo Enum (apenas sendo possível definir apartamentos do tipo SIMPLES, DUPLEX ou TRIPLEX).

A variável de instância tipoAquecimento definiu-se como tipo Enum (apenas sendo possível definir aquecimento dos tipos AR_CONDICIONADO, PISO_RADIANTE, AQUECIMENTO_CENTRAL, ou NENHUM).

Classe Loja

Variáveis de Instância

```
private int area;  
private boolean wc;  
private String tipoNegocio;  
private int numPorta;  
private int anoConstrucao;
```



A Classe Loja é uma classe que herda da Classe Imovel, e possui as variáveis de instância que herdou desta, e ainda as seguintes: tipo, areaImplantacao, areaTotalCoberta, areaTerrenoEnvolvente, numQuartos, numWCs, numPorta, anoConstrucao e tipoAquecimento.

Classe LojaComApartamento

Variáveis de Instância

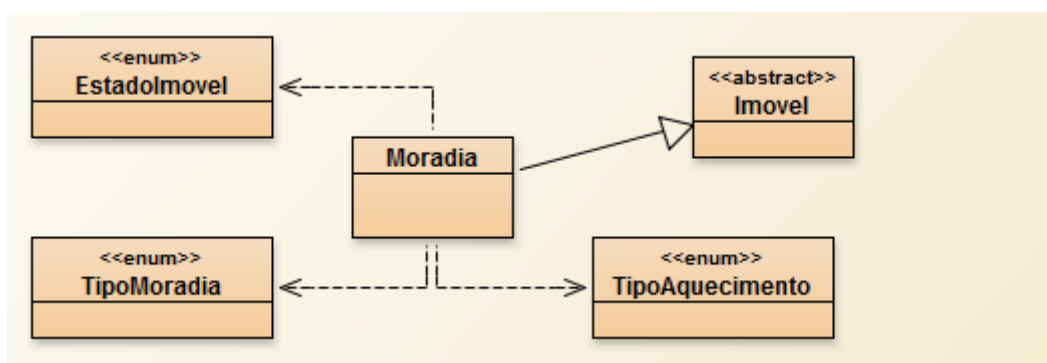
```
private Apartamento apartamento;
```

A Classe Loja com Apartamento herda da superclasse Loja. A classe Loja com Apartamento especializa a Loja, uma vez que a Loja com Apartamento possui como variável de instância um Apartamento.

Classe Moradia

Variáveis de Instância

```
private TipoMoradia tipo;  
private double areaImplantacao;  
private double areaTotalCoberta;  
private double areaTerrenoEnvolvente;  
private int numQuartos;  
private int numWCs;  
private int numPorta;  
private int anoConstrucao;  
private TipoAquecimento tipoAquecimento;
```



A Classe Moradia é uma classe que herda da Classe Imovel, e possui as variáveis de instância que herdou do desta, e ainda as seguintes: tipo, areaImplantacao, areaTotalCoberta, areaTerrenoEnvolvente, numQuartos, numWCs, numPorta, anoConstrucao e tipoAquecimento.

A variável de instância tipo definiu-se como tipo Enum (apenas sendo possível definir moradias dos tipos ISOLADA, GEMINADA, BANDA e GAVETO).

◦ Todos os Utilizadores

A estrutura de dados escolhida para guardar todos os Utilizadores foi um Map<String, Utilizador> com a chave o email do utilizador e o valor o Utilizador.

Assumiu-se que o email é único e identificativo, sendo garantido que nunca haverá emails repetidos.

Como se quer que a consulta seja rápida, e não se precisa de pesquisar utilizadores com base nas suas características, escolhemos utilizar HashMap pois permite obter um utilizador de forma direta pela chave, que é o seu email.

Nota: Como a chave do Map é uma <String> garante-se que o método hashCode() fornecido pelo Java é eficiente, salvaguardando assim uma consulta eficiente.

◦ Todos os Imóveis

A estrutura de dados escolhida para guardar todos os Imóveis é Map<String, Imovel> com a chave o código do imóvel e o valor o Imóvel.

Uma vez mais assumiu-se que código do imóvel é único e identificativo, sendo garantido que nunca haverá imóveis repetidos.

Decidiu-se usar TreeMap para guardar todos os imóveis pois eventualmente poder-se-á listar todos os imóveis presentes num sistema numa lista ordenada alfabeticamente.

Nota: Como a chave do Map é uma <String> o método compareTo() para <String> é fornecido pelo Java.

◦ Utilizador Autenticado

Como existe apenas um utilizador autenticado, este é guardado por uma variável de instância do tipo <String> com o email desse utilizador (comprador ou vendedor). Se não tiver ninguém autenticado é uma string vazia.

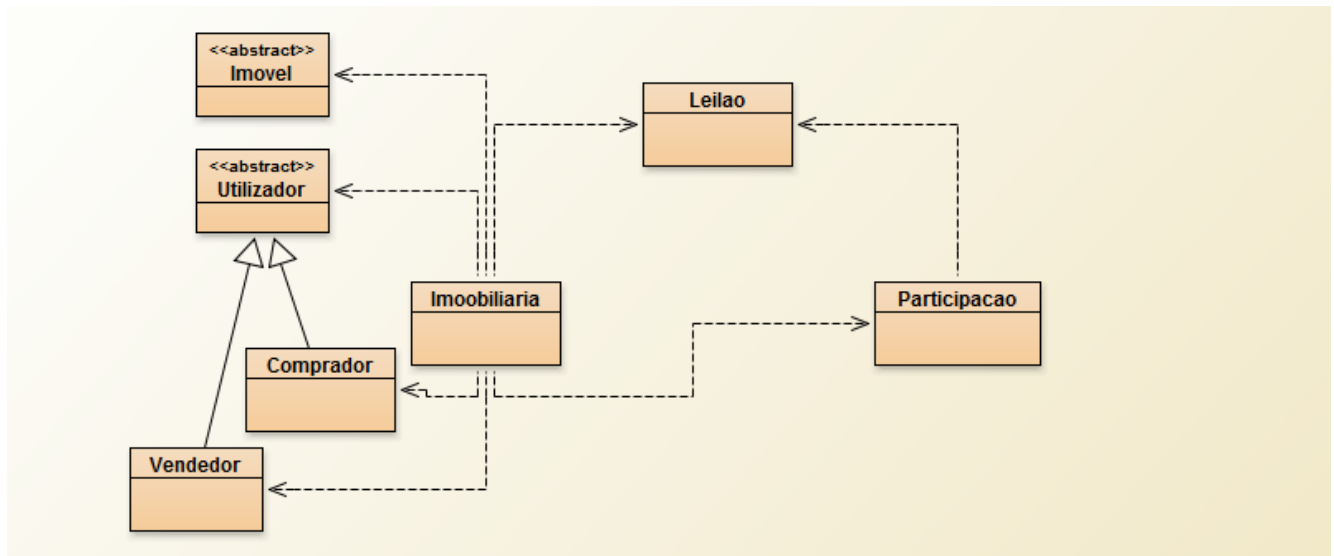
◦ Leilão

É uma das variáveis de instância desta classe, e é gerido por ela.

A interface Imobiliável contém a API que indica os métodos que uma imobiliária tem de ter para ser considerada uma imobiliária.

Variáveis de Instância

```
private String imovel;
private int duracao;
private int tempoDecorrido;
private String ultimoLicitador;
private double propostaActual;
private List<Participacao> participacoes;
```



A classe **Imobiliaria** contém uma variável **leilão** que indica o leilão em curso na aplicação. O vendedor tem a hipótese de iniciar um leilão, escolhendo um imóvel seu e uma duração.

Após o leilão estar iniciado (aberto), é possível adicionar-lhe compradores para participarem nele, definindo-lhes uma participação (valor máximo que podem oferecer, intervalo de licitações, incremento entre licitações, etc).

Após definidos os compradores a participarem no leilão, pode-se arrancar o leilão que fará uma simulação de licitações até o leilão estar terminado.

Como se poderá ver mais adiante na interface de utilizador, esta simulação vai apresentando ao comprador no ecrã, em tempo real, o estado do leilão e as licitações que forem ocorrendo.

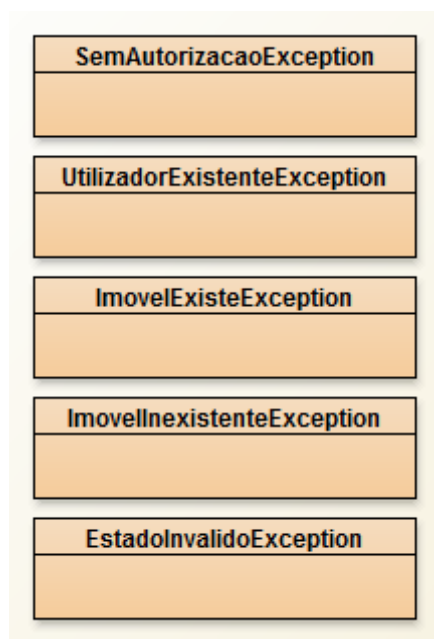
Quando o leilão termina, caso o preço mínimo seja atingido, o imóvel é marcado como reservado e é definido o Comprador associado.

Esta informação é devolvida quando se encerra o leilão.

Exceções

Neste projeto implementamos 5 exceções:

- **ImovellnexistenteException**: é lançada se o imóvel não existir.
- **UtilizadorExistenteException**: é lançada quando um utilizador deseja registar-se na imobiliária e já existe no sistema.
- **SemAutorizacaoException**: é lançada quando um utilizador não tem autorização para realizar dado método pois não está autenticado (quer como vendedor ou comprador).
- **EstadoInvalidoException**: é lançada quando o estado do imóvel não é válido, ou seja, não é um dos seguintes: VENDA, RESERVADO ou VENDIDO.
- **ImovelExisteException**: é lançada quando quero registar um imóvel na imobiliária e ele já existe no sistema.



5. Interface de Utilizador

Inicialmente, a aplicação é iniciada e pergunta:

```
Deseja carregar o ficheiro com informação previamente guardada em memória?  
1 - Sim  
2 - Não  
  
1  
  
O seu estado foi carregado com sucesso!  
  
Prima Enter para continuar  
|
```

Caso o ficheiro não exista, é exibido:

```
Deseja carregar o ficheiro com informação previamente guardada em memória?  
1 - Sim  
2 - Não  
  
1  
  
O seu estado não foi carregado. Ficheiro não encontrado.  
  
Prima Enter para continuar  
|
```

Ou caso o utilizador decida não carregar o ficheiro:

```
Deseja carregar o ficheiro com informação previamente guardada em memória?  
1 - Sim  
2 - Não  
  
2  
  
O seu estado não foi carregado.  
  
Prima Enter para continuar  
|
```

Esta aplicação para além da funcionalidade de carregar um ficheiro com a informação previamente guardada, ainda possui a possibilidade de guardar a informação do estado atual do programa num ficheiro:

```
Deseja guardar a informação num ficheiro em memória?  
1 - Sim  
2 - Não  
  
1  
  
O seu estado foi guardado com sucesso! Agradecemos a sua visita.  
  
Aplicação terminada.
```

Caso o utilizador não decida guardar a informação:

```
Deseja guardar a informação num ficheiro em memória?  
1 - Sim  
2 - Não  
  
2  
  
O seu estado não foi guardado. Agradecemos a sua visita.  
  
Aplicação terminada.
```

Página Inicial

No menu inicial, são mostrados ao utilizador os métodos iniciais, que todos os utilizadores registados ou não, podem aceder:

```
*** Seja Bem-vindo(a) à ImOObiliaria. ***

1. Iniciar sessão.

2. Registar.

3. Consultar a lista de todos os imóveis de um dado tipo e até um determinado preço.

4. Consultar todos os imóveis habitáveis até um determinado preço.

5. Consultar todos os imóveis de uma determinada tipologia (T1, T2, T3, etc).

6. Mapeamento entre imóveis e o respectivo vendedor.

7. Pesquisar um imóvel.

8. Sair do programa.

Selecione a opção que deseja.
```

Caso o utilizador insira a opção (1) é exibido no ecrã:

```
Insira o seu email:
luismanuel@imoobiliaria.pt
Insira a sua password:
34gr3rt5
```

Se o utilizador não inserir os dados corretos:

```
Insira o seu email:
luismanuel@imoobiliaria.pt
Insira a sua password:
32bji3r

As informações que introduziu não estão correctas!
```

Caso o utilizador insira a opção (2) é exibido no ecrã:

```
Deseja criar uma conta de comprador ou vendedor?
1 - Comprador
2 - Vendedor
```

De seguida é pedido ao utilizador que introduza os seus dados pessoais:

```
Insira o seu email:

pedrosoares@imoobiliaria.pt

Insira a sua password.

52g34tre

Insira o seu nome.

Pedro Soares
```



```
Insira a sua morada.  
  
Rua dos Anjos  
  
Insira a sua data de nascimento.(dd/mm/aaaa)  
  
04/06/1983  
  
A sua conta foi criada com sucesso.
```

Caso o utilizador insira a opção (3) é perguntado que tipo de imóveis deseja consultar:

```
Que tipo de imóveis deseja consultar?  
1 - Moradia  
2 - Apartamento  
3 - Loja  
4 - Terreno
```

E qual o preço máximo dos imóveis que o utilizador deseja consultar:

```
Diga qual o preço máximo dos imóveis que deseja consultar.  
150000
```

E os imóveis são devolvidos no seguinte formato:

```
.: Moradia :.  
Código: Imovel3  
Rua: Campo Grande  
Preço Pedido: 122844  
Estado: VENDA  
Tipo de moradia: GAVETO  
Area de implantacao: 187  
Area total coberta: 145  
Area de terreno envolvente: 228  
Numero de quartos: 0  
Numero de WCs: 3  
Numero de Porta: 67  
Ano de construcao: 2005  
Tipo de aquecimento: NENHUM  
Contacto vendedor: v10@vendedores.com
```

```
Prima Enter para continuar  
|
```

Caso o utilizador insira a opção (4) é perguntado o preço máximo aceitado pelo utilizador:

```
Diga qual o preço máximo dos imóveis habitáveis que deseja consultar.  
200000
```

E tal como na opção (3), se seguida são apresentados todos os imóveis com as características pedidas pelo utilizador (habitáveis e com um preço máximo determinado pelo utilizador).

Na opção (5), são mostrados ao utilizador todos os imóveis com determinada tipologia escolhida pelo utilizador (T1, T2, T3, etc):

```
Que tipo de imóveis deseja consultar?  
Exemplo: T3 possui 3 quartos.  
Insira o número de quartos:  
|
```

Selecionada a opção (6), mapeamento entre os imóveis e o respetivo vendedor, é apresentado ao utilizador:

```
O Vendedor:
    Email: v4@vendedores.com
    Nome: Nicanor Pinto
    Morada: Socorro
    Data de nascimento: 15/10/1946

Possui o Imovel:
    .: Loja :.
    Código: Imovel99
    Rua: Mercês
    Preço Pedido: 90087
    Estado: VENDA
    Área: 192
    Posui WC: Nao
    Tipo de Negócio: Mercearia
    Número da Porta: 9
    Ano Construção: 1993
    Contacto vendedor: v4@vendedores.com

Prima Enter para continuar
```

Quando o utilizador opta por pesquisar um imóvel (7), através do código do mesmo:

```
Diga qual o código do imóvel que deseja pesquisar:
Imovel99
    .: Loja :.
    Código: Imovel99
    Rua: Mercês
    Preço Pedido: 90087
    Estado: VENDA
    Área: 192
    Posui WC: Nao
    Tipo de Negócio: Mercearia
    Número da Porta: 9
    Ano Construção: 1993
    Contacto vendedor: v4@vendedores.com

Prima Enter para continuar
```

Página Inicial do Comprador

Na Página Comprador, à qual apenas um utilizador autenticado como comprador consegue aceder, é mostrado ao utilizador os métodos iniciais a que todos os utilizadores conseguem aceder, mais os métodos próprios do comprador:

```
*** Página inicial do Comprador ***
```

1. Consultar a lista de todos os imóveis de um dado tipo e até um determinado preço.
 2. Consultar todos os imóveis habitáveis até um determinado preço.
 3. Consultar todos os imóveis de uma determinada tipologia (T1, T2, T3, etc).
 4. Mapeamento entre imóveis e o respectivo vendedor.
 5. Pesquisar um imóvel.
 6. Marcar um imóvel como favorito.
 7. Consultar os imóveis favoritos.
 8. Desmarcar um imóvel como favorito.
 9. Consultar dados pessoais.
 10. Alterar dados pessoais.
 11. Terminar sessão.
- Selecione a opção que deseja.

Caso o comprador selecione a opção (6), um método próprio seu, é exibido:

```
Insira o codigo do imovel que deseja marcar como favorito:
```

```
Imovel199
```

```
O imóvel Imovel199 foi marcado como favorito com sucesso!
```

```
Prima Enter para continuar
```

```
|
```

Selecionada a opção (7), a lista dos favoritos é mostrada ao utilizador da seguinte forma:

```
O conjunto dos seus imóveis favoritos é:
```

```
.: Loja :.  
Código: Imovel199  
Rua: Mercês  
Preço Pedido: 90087  
Estado: VENDA  
Área: 192  
Posui WC: Nao  
Tipo de Negócio: Mercearia  
Número da Porta: 9  
Ano Construção: 1993  
Contacto vendedor: v4@vendedores.com
```

A opção (8) permite desmarcar um imóvel como favorito:

```
Insira o código do imóvel que deseja desmarcar como favorito:
Imovel99
O seu imóvel foi desmarcado dos favoritos com sucesso!

Prima Enter para continuar
|
```

Selecionando a opção (9) o comprador consegue visualizar os seus dados pessoais:

```
Dados pessoais:
  Email: carminhoteles@gmail.com
  Nome: Carminho Teles
  Morada: Rua dos Ferros
  Data de nascimento: 08/09/1978

- Lista de imóveis Favoritos:
  Imovel favorito nº: 1: Imovel99
  Imovel favorito nº: 2: Imovel98
```

Com a opção (10) o comprador consegue alterar os seus dados pessoais:

```
Qual o dado pessoal que deseja alterar?
```

- 1 - Nome
- 2 - Password
- 3 - Morada
- 4 - Data de nascimento (mm/dd/aaaa)

```
Insira o novo nome:
Carminho Vanessa Teles

Os seus dados pessoais foram alterados com sucesso!
```

Página Inicial do Vendedor

Na Página Inicial do Vendedor, à qual apenas um utilizador autenticado como vendedor consegue aceder, é mostrado ao utilizador os métodos iniciais a que todos os utilizadores conseguem aceder, mais os métodos próprios do vendedor:

*** Página inicial do Vendedor ***

1. Consultar a lista de todos os imóveis de um dado tipo e até um determinado preço.
 2. Consultar todos os imóveis habitáveis até um determinado preço.
 3. Consultar todos os imóveis de uma determinada tipologia (T1, T2, T3, etc).
 4. Mapeamento entre imóveis e o respectivo vendedor.
 5. Pesquisar um imóvel.
 6. Consultas (abre novo menú).
 7. Adicionar um imóvel.
 8. Alterar o estado de um imóvel para [Vendido / Reservado / Para Venda].
 9. Registrar a venda de um imóvel.
 10. Remover um imóvel.
 11. Alterar dados pessoais.
 12. Leiloar imóvel.
 13. Terminar sessão.
- Selecione a opção que deseja.

Caso o comprador selecione a opção (7), é-lhe permitido adicionar um imóvel:

*** Adicionar um Imóvel *** :

Insira o código do imovel:

Imovel17

Insira a rua do imovel:

Rua das Rosas

Insira o preço pedido do seu imovel.

128977

Insira o preço mínimo do seu imovel.

118787

Que tipo de Imovel deseja adicionar? [Moradia, Apartamento, Loja, Terreno]

1 - Moradia

2 - Apartamento

3 - Loja

4 - Terreno

Insira a area disponível:

400

Insira o tipo de Terreno

1 - Habitacao

2 - Armazem

2

```
Insira o diâmetro das canalizações:
8

Possui rede instalada?
1 - Sim
2 - Não
1

Insira a potência máxima:
13

Possui rede de esgotos?
1 - Sim
2 - Não
1
O seu imóvel foi adicionado com sucesso!
```

A opção (8) permite ao vendedor alterar o estado do seu imóvel para Vendido, Reservado ou Para Venda:

```
Qual o código do seu imóvel?
ImovelAna56
Para que estado deseja alterar o seu imóvel?
1 - Vendido
2 - Reservado
3 - Para Venda.
2
Estado do imóvel alterado com sucesso
```

Se o vendedor pretender registar a venda de um imóvel, terá de escolher a opção (9):

```
Qual o código do imóvel o qual deseja registar a venda?
ImovelAna56
A venda do seu imóvel foi registada com sucesso!
```

A opção (10) permite que o vendedor remove um imóvel:

```
Qual o código do imóvel que deseja remover?
Imovel199
Imóvel removido com sucesso!
```

Tal como na página do comprador, é possível alterar os dados pessoais, seleccionando a opção (11).

Menú de Consultas da Página Inicial do Vendedor

Na Página Inicial do Vendedor, existe uma opção (6) que abre este menú de consultas do vendedor, ao qual apenas um utilizador autenticado como vendedor consegue aceder.

*** Menú de Consultas ***

1. Consultar as 10 últimas consultas dos imóveis que tem para venda.
2. Consultar os imóveis mais consultados.
3. Consultar as estatísticas.
4. Consulta de portfolio de imóveis em venda.
5. Consulta de histórico de imóveis vendidos.
6. Voltar ao menu anterior.

Menú de Estatísticas do Menú de Consultas da Página Inicial do Vendedor

No Menú de Consultas da Página Inicial do Vendedor, existe uma opção (3) que abre este menú de estatísticas do vendedor, ao qual apenas um utilizador autenticado como vendedor consegue aceder.

*** Menú de Estatísticas ***

1. Número de imóveis vendidos.
2. Número de imóveis para venda.
3. Número de anúncios criados.
4. Número de visualizações dos anúncios.
5. Voltar ao menu anterior.

Leilão

O vendedor tem a possibilidade de iniciar um leilão de um seu imóvel. Se o preço mínimo não for atingido, o imóvel não é reservado.

```
Insira o código do imóvel que quer leiloar:
Imovel40

Insira a duracao do leilão:
15

A começar o leilão...

Última actualização: Comprador c12@compradores.com licitou 182044.38 pelo imóvel.
Última actualização: Comprador c9@compradores.com licitou 182319.38 pelo imóvel.
Última actualização: Comprador c12@compradores.com licitou 182592.38 pelo imóvel.
Última actualização: Comprador c9@compradores.com licitou 182867.38 pelo imóvel.
Última actualização: Comprador c12@compradores.com licitou 183140.38 pelo imóvel.
Última actualização: Comprador c9@compradores.com licitou 183415.38 pelo imóvel.
Última actualização: Comprador c12@compradores.com licitou 183688.38 pelo imóvel.
Não foi atingido o mínimo do imóvel, o imóvel não foi vendido.

Prima Enter para continuar
```

Quando a duração do leilão acabar, e caso o preço mínimo seja atingido, é exibido no ecrã a informação do comprador que ganhou o leilão.

```
Insira o código do imóvel que quer leiloar:
Imovel8

Insira a duracao do leilão:
35

A começar o leilão...

Última actualização: Comprador c12@compradores.com licitou 102943.7 pelo imóvel.
Última actualização: Comprador c7@compradores.com licitou 103076.7 pelo imóvel.
Última actualização: Comprador c4@compradores.com licitou 103202.7 pelo imóvel.
Última actualização: Comprador c12@compradores.com licitou 103378.7 pelo imóvel.
Última actualização: Comprador c7@compradores.com licitou 103511.7 pelo imóvel.
Última actualização: Comprador c12@compradores.com licitou 103687.7 pelo imóvel.
Última actualização: Comprador c7@compradores.com licitou 103820.7 pelo imóvel.
Última actualização: Comprador c4@compradores.com licitou 103946.7 pelo imóvel.
Última actualização: Comprador c7@compradores.com licitou 104079.7 pelo imóvel.
Última actualização: Comprador c12@compradores.com licitou 104255.7 pelo imóvel.
Última actualização: Comprador c7@compradores.com licitou 104388.7 pelo imóvel.
Última actualização: Comprador c4@compradores.com licitou 104514.7 pelo imóvel.
Última actualização: Comprador c12@compradores.com licitou 104690.7 pelo imóvel.
Última actualização: Comprador c7@compradores.com licitou 104823.7 pelo imóvel.
Última actualização: Comprador c12@compradores.com licitou 104999.7 pelo imóvel.
Última actualização: Comprador c7@compradores.com licitou 105132.7 pelo imóvel.
O utilizador que venceu o leilão foi o:

Email: c7@compradores.com
Nome: Alfredo Negrao
Morada: São Nicolau
Data de nascimento: 14/12/1964

Prima Enter para continuar
```


6. Discussão – Novos tipos de imóveis na aplicação

Discussão sobre como seria possível incluir novos tipos de imóveis na aplicação.

A classe genérica Imóvel permite ter diferentes extensões através do mecanismo de herança. Na aplicação foram implementados os tipos concretos de Imóvel: Moradia, Apartamento, Terreno, Loja e LojacomApartamento.

Desta forma, a classe abstrata Imóvel cria uma abstração de um imóvel e, portanto, não é preciso ter uma verdadeira noção das implementações concretas dos tipos de imóveis. Assim, tira-se partido do polimorfismo presente em Java.

Para definir um novo tipo de imóvel aproveitar-se-ia este mecanismo. Portanto, bastaria que este passasse a herdar da classe Imóvel tornando-se assim, sua subclasse.

Neste caso, teria que se implementar os métodos abstratos definidos na classe Imovel, o que permitiria definir-se como Imóvel.

O novo tipo de imóvel adicionado poderia passar a usar os métodos de imóvel já existentes sem problemas de compatibilidade e teria apenas de implementar métodos específicos do seu tipo (se necessário).

Por exemplo, ao adicionar o tipo de imóvel Escritório este passaria a ser uma classe que teria como superclasse a classe Imovel. Esta nova classe implementaria o método clone(), para poder pertencer à classe Imovel e definir-se como um tipo concreto.

A classe Imóvel passaria a ter 6 subclasses. Todas as subclasses podem utilizar os métodos que herdaram de Imóvel, comuns a todos os tipos de imóvel. Por exemplo, passaria a ser possível à classe Escritorio saber a rua onde se localiza.