



Desafio POO

CORRETORA IMOBILIÁRIA

Autor e Desenvolvedor

Gabriela Zala Coutinho Arruda

17/10/2025

Sumário

1. Objetivo do Projeto	3
2. Funcionalidades Implementadas	3
3. Estrutura do Projeto	4
4. Pontos de Melhoria Futuros	5

1. Objetivo do Projeto

Desenvolver um sistema simples em Java para execução via console, com o objetivo de simular o gerenciamento básico de aluguel de imóveis (casas, apartamentos e comércios). O foco principal é a aplicação prática dos conceitos fundamentais da Programação Orientada a Objetos (POO): Herança, Polimorfismo e Encapsulamento.

2. Funcionalidades Implementadas

O sistema oferece as seguintes funcionalidades:

- Cadastro de Imóveis:
 - Permite registrar novos imóveis dos tipos Casa, Apartamento ou Comércio.
 - Coleta informações como: Endereço, Número, Dados do Proprietário (Nome, Telefone, CPF), Valor do Aluguel Mensal, Taxa de Manutenção e se permite Pets.
 - Impede o cadastro de imóveis com o mesmo endereço e número.
- Aluguel de Imóveis:
 - Permite marcar um imóvel cadastrado como "alugado".
 - Registra os dados do Inquilino (Nome, Telefone, CPF) associado ao aluguel.
 - Exibe uma confirmação com os detalhes do contrato (Imóvel, Inquilino, Período) e os valores calculados (mensal e total com descontos, se aplicável).
- Disponibilização de Imóveis:
 - Permite marcar um imóvel previamente alugado como "disponível" novamente.
- Cálculo de Aluguel:
 - Calcula o valor total do aluguel para um determinado período (em meses).
 - O cálculo atual considera: $(\text{Valor Aluguel} + \text{Taxa Manutenção}) * \text{Meses}$.
 - Aplica descontos progressivos para contratos de 12, 24 ou 36 meses.
- Listagem de Imóveis:
 - Exibe uma lista de todos os imóveis cadastrados, mostrando seu status (disponível/alugado) e detalhes adicionais (Pets, Proprietário).

- Exibe uma lista dos imóveis que estão alugados, mostrando o nome do inquilino.
- Deleção de Imóveis:
 - Permite remover um imóvel do sistema, somente se ele não estiver alugado.
- Interface de Console:
 - Toda a interação ocorre via menu de texto no console.
 - Inclui mensagens e pausas ("Pressione Enter para continuar") para facilitar o uso.

Figura 1 – Menu Console

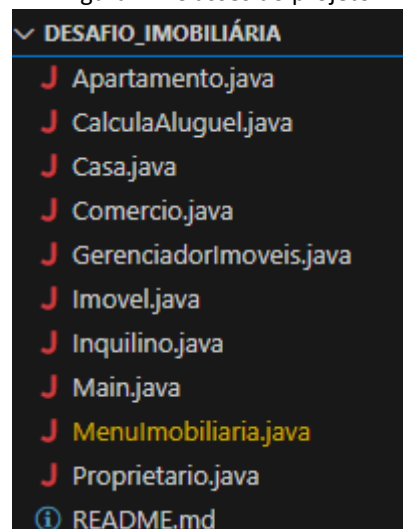
```
--- MENU PRINCIPAL ---
[1]. Cadastrar Imovel
[2]. Alugar Imovel
[3]. Disponibilizar Imovel (Encerrar Contrato)
[4]. Calcular Valor do Aluguel
[5]. Listar TODOS os Imoveis
[6]. Listar Imoveis ALUGADOS
[7]. Deletar Imovel
[0]. Sair
Digite sua opcao: █
```

Autor próprio. VS Code, 2025.

3. Estrutura do Projeto

O projeto foi desenvolvido com foco na clareza e separação de responsabilidades, utilizando as seguintes classes Java, como mostra a figura 2.

Figura 2 – Classes do projeto



Autor próprio. Vs Code, 2025

- Imovel.java: Classe abstrata base (Herança). Define atributos e métodos comuns.
- Casa.java, Apartamento.java, Comercio.java: Subclasses concretas (Herança). Implementam verificarDisponibilidade() (Polimorfismo).
- Proprietario.java, Inquilino.java: Classes de dados (Composição). Guardam informações de proprietário e inquilino.
- GerenciadorImoveis.java: Lógica de negócio e lista em memória. Gerencia cadastro, aluguel, cálculo, etc.
- CalculaAluguel.java: Classe auxiliar. Agrupa os resultados do cálculo de aluguel.
- MenuImobiliaria.java: Interface do console. Interage com o usuário (exibe menu, lê dados).
- Main.java: Ponto de entrada. Inicializa e conecta as classes.

4. Pontos de Melhoria Futuros

- Aplicar Princípios SOLID: Refatorar para melhor separação de responsabilidades.
- Identificação Única (ID): Usar IDs numéricos em vez de endereço/número para identificar imóveis. Assim o usuário irá digitar o ID quando for alugar.
- Cálculos Mais Completos: Introduzir taxas (condomínio, caução, manutenção).
- Previsão de Disponibilidade: Adicionar cálculo e exibição de data de término do contrato.