

DESAFIO POO

1 Desafio: Corretora Imobiliária

2  Objetivo

Desenvolver um sistema simples de aluguel de casas e apartamentos no console, aplicando os três pilares da POO e raciocínio lógico.

3  Requisitos

1. Classe Abstrata Imovel

o Atributos:

- Endereço
- Número
- Alugado
- Proprietário

o Métodos:

- bool estaAlugado()
- string contatoProprietário()
- int calcularAluguel(int)

2. Encapsulamento

o Endereço, número e alugado devem ser protected.

o Criar métodos públicos para obter e alterar valores de forma controlada.

3. Herança

o Criar classes:

- Casa (herda de imovel)
- Apartamento (herda de imovel)

4. Composição

o Criar classe Proprietário com atributos:

- Nome
- Telefone
- CPF

5. Polimorfismo

o Sobrescrever o método estaAlugado nas subclasses:

- Casa: retornar “A casa está disponível” ou “A casa está alugada”.
- Apartamento: retornar “O apartamento de número (número do apartamento) está disponível” ou “O apartamento de número (número do apartamento) está alugado”.

6. Lógica

o Permitir ao usuário cadastrar e deletar casa ou apartamentos.

o Permitir que seja feito o cálculo do valor total do aluguel em um determinado período de tempo.

o Permitir o usuário alugar ou disponibilizar a casa ou apartamento para ser alugado.

o Exibir no console os resultados de todas as ações do usuário.

4 ✖ Dicas para ampliar o desafio (caso queira evoluir com os mais experientes)

- Adicionar uma classe inquilino com as informações da pessoa que está alugando o imóvel.
- Criar uma lista de imóveis alugados puxando todos os imóveis que estão alugados e suas informações.
- Adicionar descontos no caso de aluguel de longa data, por exemplo 1,2 ou 3 anos.

5 ☒ O que será avaliado nesse desafio

Conceito: O que observar

Herança: Classes filhas estendendo Imovel corretamente

Polimorfismo: Sobrescrita dos método estaAlugado

Encapsulamento: Uso de modificadores (protected, private) e métodos públicos controlados

Lógica: Estrutura das ações (criação/ deleção de objetos, retornos para o usuário)

Organização: Estrutura do código, clareza e boas práticas

Criatividade: Incrementos ou variações além do mínimo