

**Integrantes:** Caio Silva Nestlehner, Igor Pereira Marani, Milena Cauana Rejani Eduardo

**Informações:**

- IDE: IntelliJ
- Nome de classes: Classe\_CaiolgorMilena
- variáveis: como estão no diagrama de classes;
- métodos: como estão no diagrama de classes;

**Instruções de uso:**

O Sistema funciona por meio de um menu de escolhas que é apresentado ao usuário. Cada escolha do usuário permite uma alteração, ou seja, cada opção invoca um determinado método.

```
"D:\Program Files (x86)\Java\jdk1.8.0_251\bin\java.exe" ...

===== SISTEMA IMOBILIARIA CAIOIGORMILENA=====
Bem-vindo!

Escolha uma opção:
1) Cadastro de Usuário;
2) Remover usuário;
3) Cadastro de Imóveis;
4) Vender Imóvel;
5) Alugar Imóvel;
6) Cadastro de Seguros;
7) Acessar relatórios;
-1) Encerrar sistema;
Opção:
|
```

Quando o usuário selecionar, cadastro de usuário, por exemplo, abrirá outro menu com opções e por fim, chegará nos métodos de cadastro, que necessitam da entrada de dados, desse modo, será requerido do usuário as informações para serem aplicadas aos métodos

```
--- Usuário a ser cadastrado:
1) Corretor;
2) Cliente;
3) Voltar ao menu;
-----

1
CADASTRO DE CORRETOR:
-- Forneça as informações a seguir:
Código de Usuário:
|
```

No exemplo da imagem, a opção do usuário foi cadastrar um corretor, então o sistema iniciou o requerimento de informações pelo código, e assim segue até cadastrar o usuário, caso ele não esteja cadastrado.

A opção “Acessar relatórios” opera diferente, uma vez que ela apenas exibe as listas armazenadas. Ao selecioná-la, o usuário escolherá a lista que deseja ver e ela aparecerá na tela caso ela exista.

```
-- Selecione o relatório que deseja visualizar: --
1) Imóveis cadastrados;
2) Casas Residenciais cadastradas;
3) Prédios Residenciais cadastrados;
4) Comerciais cadastrados;
5) Imóveis disponíveis para locação;
6) Imóveis disponíveis para venda;
7) Imóveis alugados;
8) Imóveis vendidos;
9) Imóveis com atraso no aluguel;
10) Imóveis já alugados por um Cliente em específico;
11) Imóveis já comprados por um Cliente em específico;
12) Corretores cadastrados;
13) Clientes cadastrados;
14) Clientes com alugueis em atraso;
15) Alugueis Finalizados;
16) Alugueis dentro do prazo de locação;
17) Vendas realizadas;
18) Vendas realizadas em um mês em específico e o total de lucro gerado no mês;
19) Seguros disponíveis;
20) Voltar ao Menu Principal;
```

```
Nome: Caio Silva Nestlehner
Código de usuário: 1234
CPF: 5213265587
RG: 243356698
Data de Nascimento: 2006-06-13
Endereço: Rua Unesp, 230 - CEP: 20135669
Telefone: 11935246874
E-mail: caio@unesp
Creci: 12356
Salario: 1.0E7
Pis: 12000
Data de Admissao: 2025-10-18
```

No caso acima, ao optar por exibir a lista de corretores, tivemos como saída o único corretor cadastrado e todas as suas informações.

Ao escolher sair do sistema ele finaliza a sua execução e salva os dados das listas em arquivos de texto.

### Alterações que foram feitas no diagrama de classes:

- 1) Na classe “Imovel\_CaiolgorMilena”, foram adicionadas duas variáveis booleanas, para verificação de disponibilidade quanto a locação e a venda.

```
protected Boolean disponivelVenda; 8 usages  
protected Boolean disponivelLocacao; 5 usages
```

- 2) Foi adicionada a classe “ControladordeArquivos\_CaiolgorMilena” para a criação dos métodos de tratamento dos arquivos. Como a recuperação de todas as informações presentes no arquivo (executada **sempre** no início do programa) e a função para salvar no arquivo (executada **sempre** ao final do programa, para manter os dados armazenados atualizados).

```
public class ControladordeArquivos_CaioIgorMilena { 2 usages  & Milena Cauana  
  
    public void carregarInfdeArquivos(Imobiliaria_CaioIgorMilena imobiliaria) {...}  
  
    public void salvartudonoArquivo(Imobiliaria_CaioIgorMilena imobiliaria) {...}  
}
```

- 3) Definimos o “Controlador\_CaioIgorMilena” para organização do projeto e para manter maior legibilidade na main(), uma vez que a main, apenas possui o menu que chama os métodos implementados pelo controlador. (Na imagem abaixo está apenas uma pequena parcela da classe)

```
public class Controlador_CaioIgorMilena { 2 usages  & Milena Cauana *  
  
    // ===== RELATÓRIOS OBRIGATÓRIOS =====  
  
    //Listar todos os imóveis cadastrados  
    public void relatorioImoveis(Imobiliaria_CaioIgorMilena imobiliaria) { imobiliaria.showAllImoveis(); }  
  
    //Listar todos os imóveis: Casa Residencial  
    public void relatorioCasaResidencial(Imobiliaria_CaioIgorMilena imobiliaria) {...}  
  
    //Listar todos os Imóveis de categoria: Prédio Residencial  
    public void relatorioPredio(Imobiliaria_CaioIgorMilena imobiliaria) { imobiliaria.showAllPredio(); }
```

De modo geral, essas foram as maiores alterações do que foi proposto no diagrama. O resto do projeto segue o que foi proposto, implementamos os getters e setters das classes e também os métodos que julgamos necessários.