

# Seminário POO: Sistema Imobiliário

- Ideia inicial;
- contextualização;
- Revisão de Requisitos;
- funcionamento prático.

**Aluno:** Igor Gonçalves e Eduardo Dutra  
**Turma:** Sistemas para Internet - 02/25  
**Data:** 02/12/2025

# 1.0 - Introdução

O que o sistema possui e é capaz de fazer?

- É um sistema imobiliário capaz de gerir clientes e propriedades cadastradas.
- Possui uma interface gráfica leve, intuitiva e responsiva.
- É capaz de cadastrar clientes e propriedades, editar, deletar, associar compras e locações a determinados clientes, os clientes podem definir interesse em determinada propriedade e mais.

# 1.1 - Requisitos: Propriedade

Quais os requisitos que as propriedades deveriam ter?

1. Cada propriedade deve ter um endereço, uma descrição, um tipo (casa, apartamento, terreno), um preço de venda, preço de locação e um status (disponível, em negociação, vendido ou alugado).
2. As propriedades podem ser cadastradas, removidas, atualizadas e consultadas no sistema

# 1.2 - Requisitos: Clientes

Quais os requisitos que os clientes deveriam ter?

1. Cada cliente deve ter um nome, um numero de telefone, um endereço de e-mail e uma lista de propriedades de interesse.
2. Os clientes podem ser cadastrados, removidos, atualizados e consultados no sistema.

# 1.3 - Requisitos: Operações

Quais as operações mínimas que o sistema deve atender?

1. Implemente operações para cadastrar, listar, atualizar e consultar propriedades e clientes.
2. Permita que um cliente marque uma propriedade como de interesse.
3. Forneça capacidade de marcar uma propriedade como vendida ou alugada.

# 1.4 - Entidades

## Quais as principais entidades e objetos?

- Imobiliária
  - Classe que centraliza as regras de negócio. Mantém as listas de clientes e propriedades, conversa com o DataManager para salvar/carregar os dados e oferece métodos de cadastro, busca, atualização e remoção.
- Propriedade
  - Representa cada imóvel. Controla regras de negócio como: tipo, disponibilidade, preços e status.
- Clientes
  - Representa cada cliente do sistema, com nome, telefone, e-mail e lista de propriedades de interesse. Garante, que os dados estejam sempre consistentes.

# 2.0 - Frontend

## Como funciona o Frontend do sistema?

O frontend do sistema foi criado através da biblioteca CustomTkinter. Ela é voltada para criação de interfaces gráficas sendo considerada por muitos uma evolução do TkInter.

Através desta biblioteca nós podemos criar interfaces gráficas com extrema facilidade no Python, pois ela é leve, fácil de usar e possui uma comunidade extensa.

## 2.1 - Backend

### Como funciona o Backend do sistema?

O Backend do sistema tem diversas camadas, criado com Python puro ele é capaz de gerir e armazenar todos os dados do sistema em arquivos *JSON*.

Além dos três principais objetos que temos no projeto, existe um arquivo responsável por intermediar a conexão com os dados, podendo salvar, editar, excluir e manter os dados salvos mesmo após a reinicialização do sistema.

Ele cria uma pasta e gere dois arquivos *JSON*, neles podemos encontrar todos os dados das propriedades e dos clientes cadastrados no sistema.



# 3.0 - Funcionamento

## Vejamos como funciona o sistema na prática

Vamos compreender como o sistema funciona através da demonstração prática. Veremos durante a apresentação:

- cadastro de um cliente;
- cadastro de uma propriedade;
- marcação de interesses;
- venda e aluguel de uma propriedade;
- mudar para 'em negociação' usando o botão Negociar

# 4.0 - Conclusão

## O que o sistema cumpriu?

- O sistema atendeu aos requisitos funcionais: cadastro, consulta, edição e exclusão de clientes e propriedades, interesse e marcação de venda/locação.
- Na parte de POO, o projeto aplicou encapsulamento nas classes de domínio (Cliente, Propriedade, Imobiliária) e separação entre frontend, regras de negócio e persistência.
- O uso de JSON permite manter os dados mesmo após fechar o programa, sem precisar de um banco de dados completo.
- A interface gráfica com CustomTkinter torna o uso mais próximo de um sistema real de mercado.

# 5.0 - Referências e Links

## Links de acesso as informações e acesso ao projeto

- Site oficial do Python:  
<https://www.python.org/>
- Site oficial do JSON:  
<https://www.json.org/json-pt.html>
- Biblioteca do CustomTkinter:  
<https://customtkinter.tomschimansky.com>
- Repositório do projeto no GitHub:  
<https://github.com/igorgp06/Sistema-para-Internet-IFSC>

**Fim!**

**Obrigado pela Atenção.**