

Algorithmic Thinking & Introduction to Object- Oriented



Imobiliária R.M

Aluno: Marco Antonio Monteiro Pedro

Sumário

Visão geral do projeto	3
Desafio Técnico: Gerar um Orçamento de Aluguel Mensal	3
Contexto	3
Objetivo.....	3
Funcionalidades Requeridas.....	3
Tecnologias Utilizadas	4
Estrutura de arquivos e módulos.....	5
Classes e funções principais	5
class Imovel (imovel_model.py).....	5
class Contrato (contrato_model.py).....	6
salvar_csv() (exportador_csv.py)	6
gerar_pdf_orcamento() (app.py).....	6
Bibliotecas Utilizadas	7
Interface gráfica (Tkinter).....	7
Saídas geradas.....	7
Conclusão do Projeto — Imobiliária R.M	8
Destaques técnicos:	8
Bibliotecas aplicadas.....	8
Potencial de evolução:	8

Visão geral do projeto

Este sistema é uma aplicação Python que simula uma **imobiliária digital** chamada **Imobiliária R.M**, permitindo ao usuário calcular o valor de aluguel de um imóvel, gerar um contrato parcelado, salvar os dados em **JSON** e exportar o orçamento em **PDF**. A interface gráfica é feita com **Tkinter**, tornando o uso acessível e visual.

Desafio Técnico: Gerar um Orçamento de Aluguel Mensal

Contexto

A empresa **Imobiliária R.M** busca modernizar seu processo de atendimento ao cliente, automatizando a geração de orçamentos para locação de imóveis. A proposta é desenvolver uma aplicação digital que permita calcular o valor do aluguel mensal com base nas características do imóvel e gerar documentos que facilitem a negociação.

Objetivo

Desenvolver uma aplicação que atenda às necessidades operacionais e comerciais da Imobiliária R.M, especializada na locação de **casas, apartamentos** e **estúdios**, permitindo:

- Cálculo automático do aluguel mensal
- Simulação de contrato parcelado
- Exportação dos dados em formatos digitais

Funcionalidades Requeridas

1. Geração de Orçamento

A aplicação deve considerar as seguintes regras de negócio:

- **Tipos de imóvel e valores base:**
 - Apartamento: R\$ 700,00 (1 quarto)
 - Casa: R\$ 900,00 (1 quarto)
 - Estúdio: R\$ 1200,00
- **Contrato imobiliário:**

- Valor fixo de R\$ 2.000,00
 - Parcelado em até 5 vezes
- **Adicionais por tipo:**
 - Apartamento com 2 quartos: + R\$ 200,00
 - Casa com 2 quartos: + R\$ 250,00
 - Garagem (casa ou apartamento): + R\$ 300,00
 - Estúdio com 2 vagas: + R\$ 250,00
 - Vagas extras no estúdio: + R\$ 60,00 por vaga adicional
- **Desconto:**
 - Apartamentos sem crianças recebem 5% de desconto no aluguel
- **Resultado final:**
- Exibir valor do aluguel mensal
- Exibir valor total do contrato e parcelas
- Gerar arquivo .json com os dados
- **Desafio adicional:** Gerar arquivo .csv com as **12 parcelas mensais** do orçamento

Tecnologias Utilizadas

- **Python:** linguagem principal
- **Tkinter:** interface gráfica
- **JSON:** exportação estruturada dos dados
- **FPDF:** geração de documento PDF
- **CSV (desafio):** exportação das parcelas mensais para planilh

Estrutura de arquivos e módulos

Arquivo	Função principal
main.py	Versão terminal do app (entrada via console)
app.py	Interface gráfica com Tkinter
imovel_model.py	Lógica de cálculo do aluguel
contrato_model.py	Lógica de contrato e parcelas
exportador_json.py	Exportação dos dados para JSON
data/orçamento.json	Arquivo gerado com os dados do orçamento
data/orçamento.pdf	Arquivo PDF gerado com os dados formatados
Data/orçamento.csv	Arquivo CSV com as 12 parcelas mensais

Classes e funções principais

`class Imovel (imovel_model.py)`

Representa um imóvel e calcula o valor do aluguel com base em:

- Tipo (apartamento, casa, estúdio)
- Número de quartos
- Presença de garagem
- Presença de crianças
- Vagas (apenas para estúdios)

Fórmula: valor = base + adicionais – descontos

`class Contrato (contrato_model.py)`

Define um contrato com valor total fixo (R\$ 2000) e número de parcelas (padrão: 5).

Fórmula: $\text{valor_parcela} = \text{valor_total} / \text{parcelas}$

`salvar_json()` (`exportador_json.py`)

Salva os dados do orçamento em um arquivo JSON formatado.

Inclui:

- Tipo de imóvel
- Quartos, garagem, crianças
- Valor do aluguel
- Valor total do contrato
- Parcelas mensais combinadas

`salvar_csv()` (`exportador_csv.py`)

Gera um arquivo .csv com as 12 parcelas mensais do orçamento.

Inclui:

- Mês
- Valor do aluguel
- Parcela do contrato
- Valor total mensal

`gerar_pdf_orcamento()` (`app.py`)

Gera um arquivo PDF com os dados do orçamento, formatado para impressão.

Inclui:

- Cabeçalho

- Dados principais
- Lista de parcelas mensais
-

Bibliotecas Utilizadas

- json: serializa e salva os dados em formato .json
- fpdf: gera documentos PDF com layout personalizado
- tkinter: constrói a interface gráfica interativa
- csv: exporta os dados mensais para planilhas

Interface gráfica (Tkinter)

A interface permite:

- Selecionar tipo de imóvel
- Informar número de quartos
- Marcar garagem e crianças
- Informar vagas (se estúdio)
- Calcular orçamento
- Visualizar JSON
- Exportar PDF
- Exportar CSV

Saídas geradas

data/orcamento.json

data/orcamento.pdf

data/orcamento.CSV

Conclusão do Projeto — Imobiliária R.M

O sistema desenvolvido representa uma solução completa e funcional para simulação de orçamentos imobiliários, combinando **lógica de negócios**, **persistência de dados** e **interface amigável**. A estrutura modular com classes bem definidas permite fácil manutenção e expansão futura.

Destaques técnicos:

- **Organização por módulos:** separação clara entre cálculo de aluguel, contrato, exportação e interface.
- **Validação de entrada:** garante integridade dos dados fornecidos pelo usuário.
- **Exportação em JSON, PDF e CSV:** permite salvar e compartilhar orçamentos de forma estruturada e profissional.
- **Interface gráfica com Tkinter:** torna o sistema acessível a usuários sem conhecimento técnico.

Bibliotecas aplicadas:

- json: para persistência de dados em formato legível e reutilizável.
- fpdf: para geração de documentos PDF prontos para impressão.
- tkinter: para construção de uma interface visual intuitiva.

Potencial de evolução:

- Adição de login e histórico de orçamentos
- Integração com banco de dados
- Estilização avançada do PDF com logotipo e layout corporativo

