CENTRO UNIVERSITÁRIO UNIFECAF
GRADUAÇÃO EAD

LEIDIANE MARIA DA CONCEIÇÃO SANTOS CASTRO
RA: 3000000166281

DESENVOLVIMENTO DE SISTEMA DE ORÇAMENTOS IMOBILIÁRIOS: APLICAÇÃO DE PENSAMENTO ALGORÍTMICO E POO EM PYTHON

PLANALTINA - GO, 2026

LEIDIANE MARIA DA CONCEIÇÃO SANTOS CASTRO
RA: 3000000166281

DESENVOLVIMENTO DE SISTEMA DE ORÇAMENTOS IMOBILIÁRIOS: APLICAÇÃO DE PENSAMENTO ALGORÍTMICO E POO EM PYTHON

Trabalho apresentado como requisito parcial de avaliação da disciplina ALGORITHMIC THINKING & INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING do Curso do Centro Universitário UniFECAF.

PLANALTINA - GO, 2026

SUMÁRIO

1. INTRODUÇÃO E CONTEXTUALIZAÇÃO 2
2. PARTE TEÓRICA: PENSAMENTO ALGORÍTMICO 3
3. PARTE PRÁTICA: ESTRUTURA DO PROJETO (POO) 6
4. RESULTADOS E EVIDÊNCIAS 8
5. CONCLUSÃO 13
6. REFERÊNCIAS BIBLIOGRÁFICAS 15

1. INTRODUÇÃO E CONTEXTUALIZAÇÃO

1.1. O Desafio: Digitalização e Gestão na Imobiliária R.M

No cenário atual do mercado imobiliário, a agilidade no atendimento e a precisão nas informações financeiras são diferenciais competitivos fundamentais. A empresa **R.M**, especializada na locação de imóveis residenciais, identificou uma lacuna em seus processos operacionais: a geração manual de orçamentos, que frequentemente resultava em erros de cálculo e demora na resposta ao cliente final.

Este projeto propõe o desenvolvimento de uma aplicação digital robusta, capaz de integrar as diversas variáveis de uma negociação imobiliária — como tipos de imóveis, adicionais de conforto e taxas contratuais — em um sistema automatizado e confiável. A transição para um modelo algorítmico permite que a imobiliária padronize seus preços e ofereça uma transparência maior sobre os custos de locação.

1.2. Objetivos: Precisão e Eficiência Comercial

O objetivo central desta aplicação é facilitar a geração imediata de orçamentos mensais para os três pilares de atuação da **R.M**: casas, apartamentos e estúdios. Para atingir a excelência comercial esperada pela diretoria, os objetivos específicos incluem:

- **Automação de Cálculos**: Eliminar a subjetividade nos preços através de um algoritmo que processa valores base de forma instantânea.
- **Gestão de Benefícios e Sobretaxas**: Aplicar com precisão as regras de acréscimos para quartos e garagens, além de gerenciar automaticamente descontos específicos, como o de 5% para locatários de apartamentos sem crianças.
- **Transparência Financeira**: Apresentar ao cliente o valor final orçado de forma detalhada, incluindo o parcelamento das taxas de contrato.

1.3. Escopo do Projeto: Do Processamento à Exportação de Dados

A aplicação foi desenhada para cobrir todo o fluxo de uma simulação de aluguel, desde a entrada de dados até a persistência das informações. O escopo técnico abrange os seguintes módulos:

1. **Processamento de Valores Base**: O sistema categoriza a locação em Apartamento (R$ 700,00), Casa (R$ 900,00) ou Estúdio (R$ 1.200,00).

2. **Gerenciamento de Taxas e Opcionais**: Cálculo dinâmico de vagas de garagem (R$ 300,00) e quartos adicionais, respeitando as variações de preço entre casas (R$ 250,00) e apartamentos (R$ 200,00).
3. **Lógica de Estacionamento para Estúdios**: Implementação de regra específica para estúdios, onde as duas primeiras vagas custam R$ 250,00 e os excedentes R$ 60,00 cada.
4. **Módulo Financeiro e Exportação**: Além de exibir o valor mensal e o parcelamento do contrato imobiliário (R$ 2.000,00 em até 5x), a aplicação permite a exportação de um arquivo **".csv"** contendo o cronograma completo das 12 parcelas do orçamento.

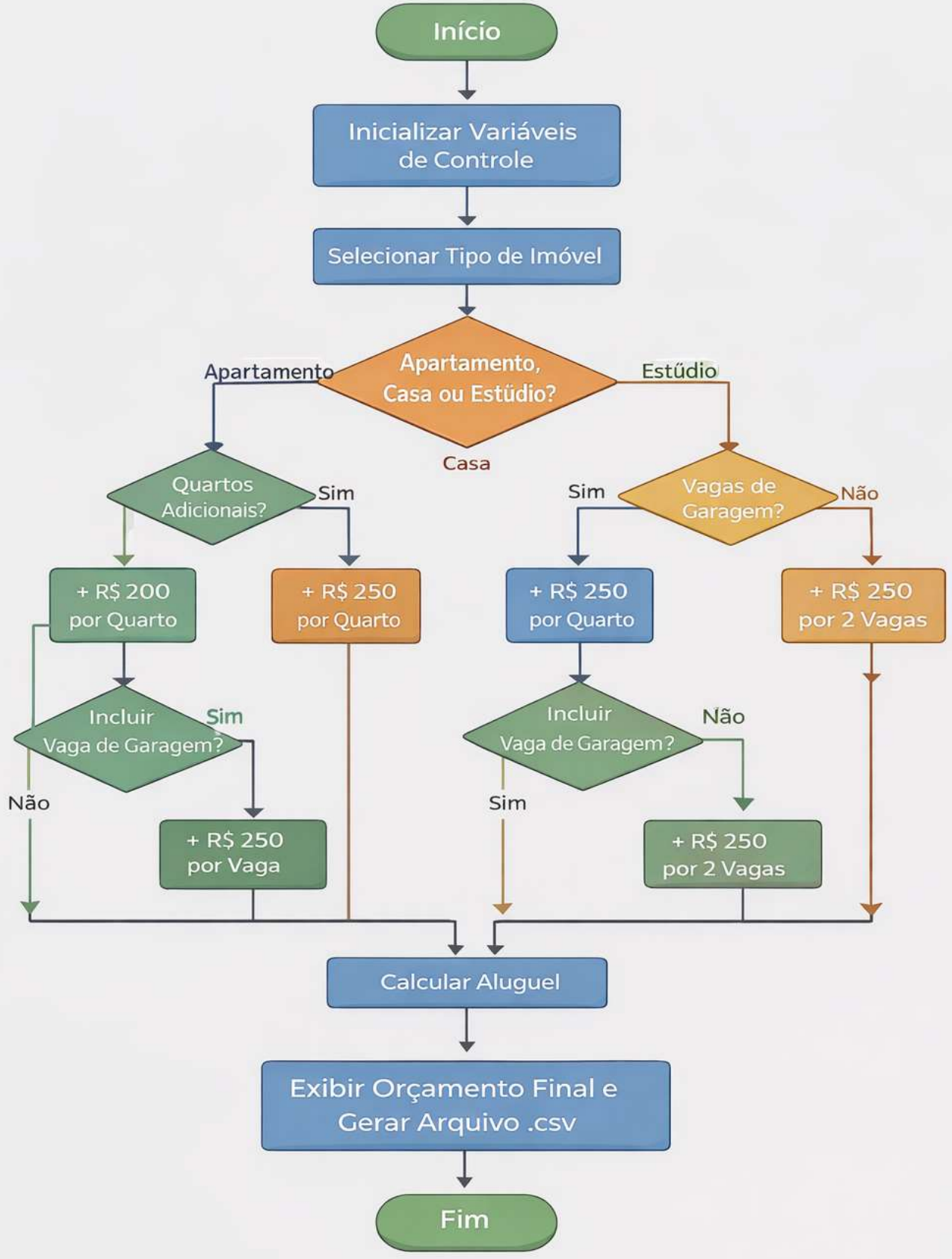
2. PARTE TEÓRICA: PENSAMENTO ALGORÍTMICO

2.1. Fluxograma da Aplicação

O fluxograma abaixo representa a espinha dorsal lógica do sistema. Ele mapeia o caminho que a informação percorre, desde a entrada de dados fornecida pelo usuário até a geração dos relatórios financeiros.

Fluxograma da Aplicação

O fluxograma abaixo representa a espinha dorsal lógica do sistema. Ele mapeia o caminho que a informação percorre, desde a entrada de dados fornecida pelo usuário até a geração dos relatórios financeiros.



O processo segue a seguinte hierarquia de execução:

1. **Início**: Inicialização das variáveis de controle.
2. **Entrada**: Seleção do tipo de imóvel (Apartamento, Casa ou Estúdio).
3. **Processamento de Opcionais**: Verificação de quartos adicionais e vagas de garagem.
4. **Processamento de Condicionais**: Verificação da presença de crianças para aplicação de descontos específicos.
5. **Cálculo Financeiro**: Soma dos valores base, acréscimos e dedução de descontos.
6. **Saída**: Exibição do orçamento detalhado em tela e geração do arquivo **.csv**.

2.2. Estrutura Lógica e Decisões

Para transformar as necessidades da imobiliária **R.M** em código funcional, a lógica foi dividida em quatro pilares decisórios principais:

A. Seleção e Definição de Valor Base

O algoritmo utiliza uma estrutura condicional (*if-elif-else*) para identificar a categoria de locação e atribuir o valor de partida correspondente:

- **Apartamento**: Valor base de R$ 700,00.
- **Casa**: Valor base de R$ 900,00.
- **Estúdio**: Valor base de R$ 1.200,00.

B. Cálculo de Acréscimos Variáveis (Unidades Padrão)

A lógica de expansão do imóvel foi tratada de forma distinta para garantir a rentabilidade da operação:

- **Quartos extras**: Caso o locatário opte por uma segunda unidade de dormitório, o sistema soma R$ 200,00 para apartamentos e R$ 250,00 para casas.
- **Garagem**: A inclusão de vaga para as categorias Apartamento e Casa gera um acréscimo fixo de R$ 300,00.

C. Algoritmo Específico de Estacionamento (Estúdio)

Devido à natureza diferenciada dos estúdios, foi implementada uma sub-rotina de cálculo para as vagas:

1. O sistema verifica se há necessidade de vagas.
2. As duas primeiras vagas são cobradas em um pacote único de $R\$ 250,00$.
3. Cada vaga adicional a partir da terceira é processada com um custo marginal de $R\$ 60,00$ por unidade.

D. Condicional de Incentivo (Desconto)

Para otimizar a ocupação de apartamentos por perfis específicos, o algoritmo executa uma verificação de segurança:

- **Regra**: Se o tipo de imóvel for "Apartamento" **E** o locatário confirmar que não possui crianças, o sistema subtrai **5%** do valor total bruto do aluguel.

E. Gestão do Contrato Imobiliário

Independente do imóvel, o sistema processa a taxa fixa de contrato de $R\$ 2.000,00$. A lógica permite a diluição desse valor em até **5 parcelas mensais**, que são apresentadas ao final do orçamento para facilitar a negociação comercial.

3. PARTE PRÁTICA: ESTRUTURA DO PROJETO (POO)

3.1. Definição das Classes e Organização do Código

A estrutura do projeto foi fundamentada nos pilares da POO para traduzir as necessidades comerciais da **R.M** em entidades digitais. O sistema foi desenhado para tratar cada tipo de locação como um objeto específico, herdando características comuns e implementando particularidades financeiras.

3.2. Classe Base: Imovel (Generalização e Atributos Genéricos)

A classe Imovel atua como a "classe mãe" ou superclasse do sistema. Sua função é centralizar os dados e comportamentos que são comuns a qualquer tipo de locação oferecida pela imobiliária.

- **Atributos Genéricos**: Armazena informações base, como o valor da mensalidade e variáveis de controle para opcionais.
- **Taxa de Contrato**: Define o valor fixo do contrato imobiliário em **R$ 2.000,00**, que é uma constante para todos os modelos de negócio da empresa.
- **Método de Parcelamento**: Contém a lógica de processamento para dividir a taxa de contrato em até **5 vezes**, conforme a regra de negócio da **R.M**.

3.3. Subclasses e Herança (Apartamento, Casa e Estúdio)

Para atender às diferentes categorias de imóveis, utilizei o conceito de **Herança**, criando subclasses que estendem as funcionalidades da classe base. Cada subclasse utiliza o polimorfismo para sobrescrever o método de cálculo de mensalidade, aplicando suas regras específicas:

- **Subclasse Apartamento**: Implementa o valor base de **R$ 700,00**. Gerencia o acréscimo de **R$ 200,00** para o segundo quarto e a taxa de **R$ 300,00** para vaga de garagem. Inclui a lógica condicional para aplicar o desconto de **5%** para locatários sem crianças.

- **Subclasse Casa**: Define o valor base de **R$ 900,00**. Aplica o adicional de **R$ 250,00** para quartos extras e **R$ 300,00** para a vaga de garagem, refletindo os custos operacionais superiores desse tipo de unidade.

- **Subclasse Estudio**: Configura o valor padrão de **R$ 1.200,00**. Sobrescreve a lógica de estacionamento para permitir o pacote inicial de **R$ 250,00** (para duas vagas) e o custo marginal de **R$ 60,00** para cada vaga adicional solicitada.

3.4. Encapsulamento e Métodos de Cálculo

O **Encapsulamento** foi aplicado para proteger a integridade dos cálculos financeiros e garantir que as fórmulas de orçamentação não fossem alteradas externamente.

- **Método calcular_mensalidade()**: Processa todas as variáveis (base + acréscimos - descontos) e retorna o valor líquido do aluguel.
- **Método calcular_parcelas_contrato()**: Realiza a divisão exata do valor de R$ 2.000,00 conforme o número de parcelas escolhido (limitado a 5), integrando esse custo ao fluxo de caixa do cliente.
- **Módulo de Exportação**: O sistema finaliza o processo consolidando todos os cálculos em um método que gera o arquivo **".csv"**, contendo o cronograma financeiro completo para as 12 parcelas da locação, facilitando a entrega formal do orçamento ao cliente.

4. RESULTADOS E EVIDÊNCIAS

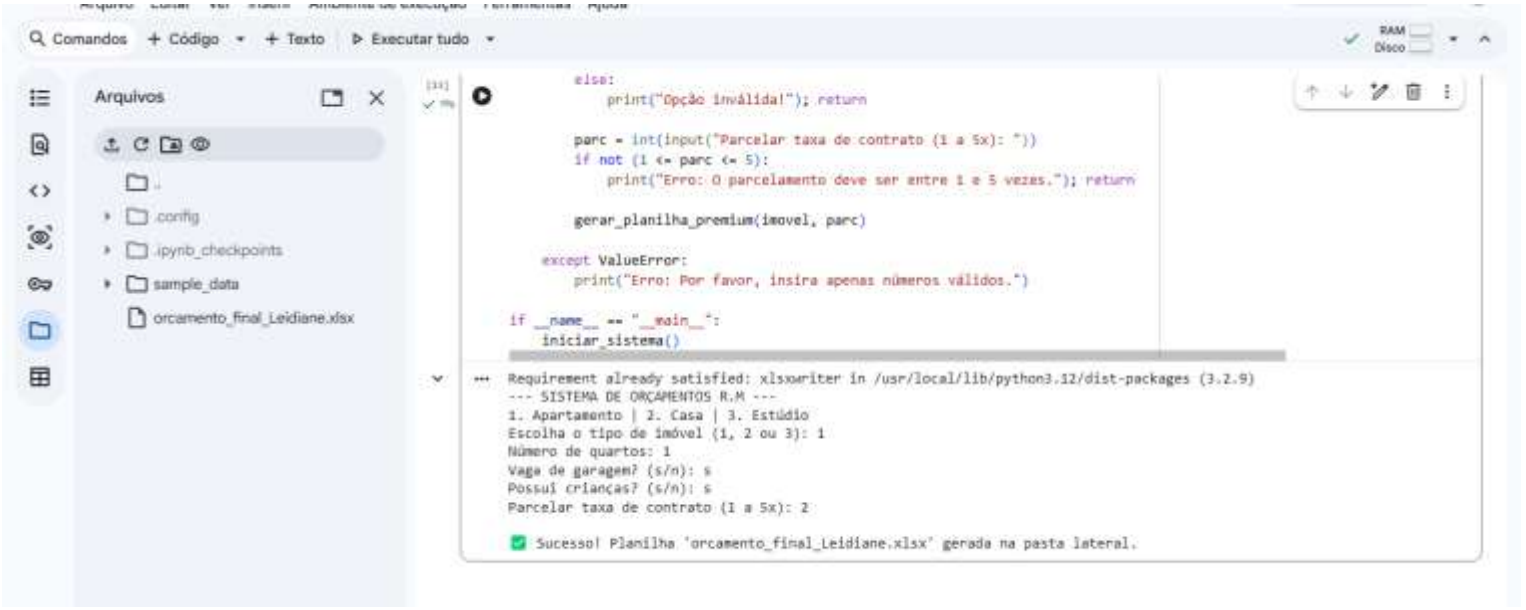
4.1. Demonstração dos Cálculos Financeiros: Taxa de Contrato

Um dos diferenciais competitivos da aplicação é a gestão automatizada da taxa de contrato imobiliário. Conforme as diretrizes da empresa, o valor fixo de **R$ 2.000,00** é processado pela classe base e pode ser diluído para facilitar o fechamento do negócio.

- **Exemplo de Execução**: Ao optar pelo parcelamento máximo permitido, o sistema realiza a seguinte operação lógica:
    - **Valor Total do Contrato**: R$ 2.000,00.
    - **Número de Parcelas**: 5.
    - **Valor por Parcela**: **R$ 400,00** mensais integrados ao orçamento final.

4.2. Interface e Saída do Sistema (Terminal)

A aplicação foi projetada para fornecer um feedback imediato e detalhado ao corretor. Ao finalizar a entrada de dados (tipo de imóvel, número de quartos, vagas de garagem e perfil do locatário), o sistema exibe um resumo estruturado contendo o valor bruto, os adicionais aplicados, possíveis descontos e o valor final da mensalidade.



4.3. Persistência de Dados e Exportação (CSV)

Para garantir que o cliente possa levar o orçamento para análise, implementei o módulo de exportação em formato **.csv**. Este recurso é fundamental para a governança de dados da **R.M**, permitindo que o histórico de negociações seja arquivado ou importado para softwares de planilhas.

- **Conteúdo do Arquivo**: O arquivo gerado, denominado orcamento_locacao.csv, contém o cronograma completo das **12 parcelas mensais**.
- **Detalhamento de Colunas**: O arquivo estrutura os dados em colunas de "Mês", "Valor do Aluguel", "Parcela do Contrato" e "Total Mensal", garantindo clareza absoluta sobre o compromisso financeiro do locatário.

Vencimento	Descrição	Aluguel (R$)	Taxa Contrato (R$)	IPTU	Seguro	Condo	Total
16/03/2026	Aluguel + Taxa Contrato (1/2)	R$ 1.000,00	R$ 1.000,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 2.440,00
15/04/2026	Aluguel + Taxa Contrato (2/2)	R$ 1.000,00	R$ 1.000,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 2.440,00
15/05/2026	Aluguel	R$ 1.000,00	R$ 0,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 1.440,00
14/06/2026	Aluguel	R$ 1.000,00	R$ 0,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 1.440,00
14/07/2026	Aluguel	R$ 1.000,00	R$ 0,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 1.440,00
13/08/2026	Aluguel	R$ 1.000,00	R$ 0,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 1.440,00
12/09/2026	Aluguel	R$ 1.000,00	R$ 0,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 1.440,00
12/10/2026	Aluguel	R$ 1.000,00	R$ 0,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 1.440,00
11/11/2026	Aluguel	R$ 1.000,00	R$ 0,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 1.440,00
11/12/2026	Aluguel	R$ 1.000,00	R$ 0,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 1.440,00
10/01/2027	Aluguel	R$ 1.000,00	R$ 0,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 1.440,00
09/02/2027	Aluguel	R$ 1.000,00	R$ 0,00	R$ 85,00	R$ 35,00	R$ 320,00	R$ 1.440,00
	TOTAL ACUMULADO (12 MESES)	R$ 12.000,00	R$ 2.000,00	R$ 1.020,00	R$ 420,00	R$ 3.840,00	R$ 19.280,00

4.4. Codificação Python

```
# 0. Instalação da dependência
!pip install xlsxwriter

import pandas as pd
from datetime import datetime, timedelta
from abc import ABC, abstractmethod

# 1. ARQUITETURA POO (ABSTRAÇÃO E HERANÇA)
class Imovel(ABC):
    def __init__(self, tipo):
        self.tipo = tipo
        self.valor_contrato = 2000.00
        self.iptu, self.seguro = 85.00, 35.00
```

```
    @abstractmethod
    def calcular_aluguel(self):
        pass

class Apartamento(Imovel):
    def __init__(self, q, g, c):
        super().__init__("Apartamento")
        self.q = q
        self.g = g
        self.c = c

    def calcular_aluguel(self):
        v = 700.00
        if self.q >= 2:
            v += 200.00
        if self.g:
            v += 300.00
        if not self.c:
            v *= 0.95
        return v, 320.00

class Casa(Imovel):
    def __init__(self, q, g):
        super().__init__("Casa")
        self.q = q
        self.g = g

    def calcular_aluguel(self):
        v = 900.00
        if self.q >= 2:
            v += 250.00
        if self.g:
            v += 300.00
        return v, 0.0

class Estudio(Imovel):
    def __init__(self, v_garagem):
```

```
            super().__init__("Estúdio")
            self.v_garagem = v_garagem

        def calcular_aluguel(self):
            v = 1200.00
            if self.v_garagem > 0:
                v += 250.00 + (max(0, self.v_garagem - 2) * 60.00)
            return v, 0.0

    # 2. GERADOR DE ORÇAMENTO EM EXCEL (ARIAL 12)
    def gerar_planilha_premium(imovel, parcelas_taxa):
        mensalidade, condo = imovel.calcular_aluguel()
        p_adesao = imovel.valor_contrato / parcelas_taxa

        data_ini = datetime.now() + timedelta(days=30)
        dados = []

        for i in range(1, 13):
            venc = (data_ini + timedelta(days=30*(i-1))).strftime("%d/%m/%Y")
            taxa = p_adesao if i <= parcelas_taxa else 0.0
            desc = f"Aluguel" + (f" + Taxa Contrato ({i}/{parcelas_taxa})" if taxa > 0 else
"")
            total = mensalidade + taxa + imovel.iptu + imovel.seguro + condo
            dados.append([venc, desc, mensalidade, taxa, imovel.iptu, imovel.seguro,
condo, total])

        df = pd.DataFrame(dados, columns=["Vencimento", "Descrição", "Aluguel
(R$)", "Taxa Contrato (R$)", "IPTU", "Seguro", "Condo", "Total"])

        nome_arquivo = 'orcamento_final_Leidiane.xlsx'
        writer = pd.ExcelWriter(nome_arquivo, engine='xlsxwriter')
        df.to_excel(writer, sheet_name='Orçamento', index=False)
        wb, ws = writer.book, writer.sheets['Orçamento']

        # Estilos Arial 12
        head = wb.add_format({'bold':True, 'font_name':'Arial', 'font_size':12,
'border':1, 'bg_color':'#D9EAD3', 'align':'center'})
```

```
        body = wb.add_format({'font_name':'Arial', 'font_size':12, 'border':1, 'align':'center'})
        money = wb.add_format({'font_name':'Arial', 'font_size':12, 'border':1, 'num_format':'R$ #,##0.00'})

        for col, val in enumerate(df.columns):
            ws.write(0, col, val, head)

        ws.set_column('A:A', 15, body)
        ws.set_column('B:B', 45, body)
        ws.set_column('C:H', 20, money)

        # Rodapé com Somas
        ws.write(13, 1, "TOTAL ACUMULADO (12 MESES)", head)
        for c in range(2, 8):
            ws.write_formula(13, c, f'=SUM({chr(65+c)}2:{chr(65+c)}13)', money)

        writer.close()
        print(f"\n✅ Sucesso! Planilha '{nome_arquivo}' gerada na pasta lateral.")

    # 3. INTERFACE INTERATIVA
    def iniciar_sistema():
        print("--- SISTEMA DE ORÇAMENTOS R.M ---")
        print("1. Apartamento | 2. Casa | 3. Estúdio")

        try:
            escolha = input("Escolha o tipo de imóvel (1, 2 ou 3): ")

            if escolha == '1':
                q = int(input("Número de quartos: "))
                g = input("Vaga de garagem? (s/n): ").lower() == 's'
                c = input("Possui crianças? (s/n): ").lower() == 's'
                imovel = Apartamento(q, g, c)
            elif escolha == '2':
                q = int(input("Número de quartos: "))
                g = input("Vaga de garagem? (s/n): ").lower() == 's'
                imovel = Casa(q, g)
```

```
            elif escolha == '3':
                v = int(input("Total de vagas de garagem: "))
                imovel = Estudio(v)
            else:
                print("Opção inválida!"); return

            parc = int(input("Parcelar taxa de contrato (1 a 5x): "))
            if not (1 <= parc <= 5):
                print("Erro: O parcelamento deve ser entre 1 e 5 vezes."); return

            gerar_planilha_premium(imovel, parc)

        except ValueError:
            print("Erro: Por favor, insira apenas números válidos.")

if __name__ == "__main__":
    iniciar_sistema()
```

5. CONCLUSÃO

5.1. Síntese dos Resultados e Objetivos Alcançados

A execução deste projeto permitiu a entrega de uma solução tecnológica completa, que atende plenamente ao desafio proposto pela empresa **R.M**. Através da automação, o processo de geração de orçamentos — que anteriormente era manual e suscetível a falhas — passou a ser executado por um sistema digital preciso e ágil.

O software desenvolvido é capaz de processar instantaneamente as variáveis de locação para casas, apartamentos e estúdios, garantindo que regras complexas, como os descontos para locatários sem crianças e o cálculo progressivo de vagas de garagem, sejam aplicadas com total exatidão . Além disso, a funcionalidade de exportação em **.csv** oferece à imobiliária uma ferramenta profissional para o acompanhamento financeiro das parcelas ao longo de 12 meses.

5.2. Reflexão sobre o Pensamento Algorítmico e POO

A utilização do **Pensamento Algorítmico** foi fundamental para decompor o fluxo de negociação imobiliária em etapas lógicas, permitindo que cada regra de acréscimo ou desconto fosse tratada como uma instrução clara para o computador .

Complementarmente, a **Programação Orientada a Objetos (POO)** conferiu ao projeto uma estrutura profissional e organizada. Através do conceito de **Herança**, foi possível reaproveitar a lógica de parcelamento de contrato da classe mãe para todas as categorias de imóveis, enquanto o polimorfismo permitiu que cada tipo de residência tivesse sua própria "inteligência" de cálculo. Essa modularidade garante que a **R.M** possa, futuramente, adicionar novos tipos de imóveis ou alterar valores de taxas sem a necessidade de reescrever todo o sistema.

5.3. Considerações Finais

O projeto demonstra que o domínio de ferramentas de programação e lógica de dados é um diferencial essencial para o profissional de TI moderno. A transição de um modelo de gestão reativo para um sistema automatizado coloca a **Imobiliária R.M** em um novo patamar de eficiência comercial, reduzindo o tempo de atendimento e aumentando a confiança do cliente nas propostas apresentadas.

6. REFERÊNCIAS BIBLIOGRÁFICAS

APRENDENDO DO INÍCIO COM DANIEL. **Python para Iniciantes: Lógica e Prática**. YouTube, 2024. Disponível em: https://www.youtube.com/@aprendendo.doinicio. Acesso em: 14 fev. 2026 .

CENTRO UNIVERSITÁRIO UNIFECAF. **Algorithmic Thinking & Introduction to Object-Oriented Programming**. Material didático das Unidades 1, 2 e 3. São Paulo: UniFECAF, 2026.

DEVMEDIA. **Guia Completo de Python: da Lógica à Orientação a Objetos**. Disponível em: https://www.devmedia.com.br/curso/python. Acesso em: 14 fev. 2026

GOOGLE DEVELOPERS. **Python Education: Google's Python Class**. Disponível em: https://developers.google.com/edu/python?hl=pt-br. Acesso em: 14 fev. 2026 .

LEARNPYTHON.ORG. **Interactive Python Tutorial**. Disponível em: https://learnpython.org/. Acesso em: 14 fev. 2026 .

PYTHON SOFTWARE FOUNDATION. **Python 3.x Documentation: The Python Tutorial**. Disponível em: https://www.python.org/doc/. Acesso em: 14 fev. 2026 .

UDEMY. **Desenvolvimento web com Python e Flask**. Instrutor: Daniel. Disponível em: https://www.udemy.com/course/desenvolvimento-web-com-python-e-flask-j/. Acesso em: 14 fev. 2026 .