



UNIVERSIDADE FEDERAL DO PARANÁ

LEONARDO DE SOUZA BERALDI
LUIZA GABRIELA DA SILVA
MATEUS MUZZILLO COELHO
MATHEUS OKASAKI TONIETO

IMOBILIÁRIA MZZ

CURITIBA

2019

LEONARDO DE SOUZA BERALDI (GRR20195849)

LUIZA GABRIELA DA SILVA (GRR20195853)

MATEUS MUZZILLO COELHO (GRR20195862)

MATHEUS OKASAKI TONIETO (GRR20195863)

IMOBILIÁRIA MZZ

Trabalho apresentado como requisito à
obtenção da aprovação na matéria de
Fundamentos da Programação de
Computadores, Setor de Ciências Exatas,
Universidade Federal do Paraná.

Orientador: Prof. Jackson Antonio do Prado Lima.

CURITIBA

2019

RESUMO

Este projeto pretende facilitar a burocracia de uma imobiliária, que é responsável pela compra e venda de imóveis residenciais, uma vez que, como há muitas fichas para serem avaliadas, tal processo acaba sendo muito demorado e pode acabar incompleto. Visando tal praticidade, criamos um programa feito em Python onde o proprietário que deseja vender seu imóvel, informe seus dados e as características de sua propriedade no programa, e aguarda a resposta da imobiliária, esta, por sua vez, tem todas as informações necessárias para aceitar ou não, e com isso há uma ordem muito mais fácil a ser seguida sem aquele monte de papel em cima das mesas dos funcionários.

LISTA DE FIGURAS

| | |
|--|----|
| FIGURA 1 – TELA PRINCIPAL – PARTE 2..... | 8 |
| FIGURA 2 – TELA PRINCIPAL – PARTE 2..... | 8 |
| FIGURA 3 – TELA PRINCIPAL – PARTE 3..... | 9 |
| FIGURA 4 – FUNÇÃO CATALOGO – PARTE 1..... | 9 |
| FIGURA 5– FUNÇÃO CATALOGO – PARTE 2..... | 9 |
| FIGURA 6– FUNÇÃO CATALOGO – PARTE 3..... | 10 |
| FIGURA 7– FUNÇÃO CATALOGO – PARTE 4..... | 10 |
| FIGURA 8– FUNÇÃO SALVANDO – PARTE 1..... | 11 |
| FIGURA 9– FUNÇÃO SALVANDO – PARTE 2 | 11 |
| FIGURA 10– FUNÇÃO PROPOSTAS – PARTE 1..... | 12 |
| FIGURA 11– FUNÇÃO PROPOSTAS – PARTE 2..... | 12 |
| FIGURA 12– FUNÇÃO ACEITAR– PARTE 1..... | 13 |
| FIGURA 13– FUNÇÃO ACEITAR– PARTE 2..... | 13 |
| FIGURA 14– FUNÇÃO NEGAR– PARTE 1..... | 14 |
| FIGURA 15– FUNÇÃO NEGAR– PARTE 2..... | 14 |
| FIGURA 16 – FUNÇÃO INSTRUÇÕES..... | 15 |
| FIGURA 17– FUNÇÃO PROCURAR – PARTE 1..... | 15 |
| FIGURA 18– FUNÇÃO INSTRUCOES – PARTE 2..... | 16 |
| FIGURA 19 – FUNÇÃO PROCURAR – PARTE 1..... | 16 |
| FIGURA 20– FUNÇÃO PROXIMA_CASA – PARTE 1..... | 17 |
| FIGURA 21 – FUNÇÃO PROXIMA_CASA – PARTE 2..... | 17 |
| FIGURA 22 – FUNÇÃO PROCURAR – PARTE 3..... | 18 |

SUMÁRIO

| | |
|---|-----------|
| 1. INTRODUÇÃO..... | 6 |
| 1.1 MOTIVAÇÃO..... | 6 |
| 1.2 OBJETIVO..... | 6 |
| 2. METODOLOGIA..... | 7 |
| 2.1 BIBLIOTECAS UTILIZADAS..... | 7 |
| 2.2 FUNÇÕES CRIADAS..... | 7 |
| 2.3 FUNCIONAMENTO DO PROGRAMA..... | 8 |
| 2.4 RESULTADOS OBTIDOS..... | 18 |
| 2.5 DIFICULDADES ENCONTRADAS..... | 18 |
| 2.6 SUGESTÕES PARA TRABALHOS POSTERIORES..... | 19 |
| 3. REFERÊNCIAS..... | 20 |

1. INTRODUÇÃO

Para que um imóvel seja comprado por uma imobiliária, existe um processo chamado de Avaliação Imobiliária, que dentre outras funções, avalia as características da propriedade e negocia com o proprietário. O processo de avaliação é feito por um perito imobiliário, que deve ter formação na área específica e ser inscrito nos conselhos profissionais regulamentadores como o Conselho Regional de Engenharia e Agronomia (CREA) e o Conselho Regional de Corretores de Imóveis (CRECI). Com isso, o programa deste relatório vem com a proposta de automatizar o trabalho deles, já que a entrada das informações necessárias estão em uma página só, e a negociação pode ser muito mais fácil, bem como colocar o imóvel a venda em apenas um clique.

1.1 MOTIVAÇÃO

Nossa motivação foi a demora para receber respostas de imobiliárias por conta do manuseio de muitas fichas, acarretando a perda de muitas oportunidades de negócio.

1.2 OBJETIVO

Com o acesso a um programa desenvolvido pelos alunos deste projeto, visamos melhorar a fluidez das demandas de pedidos nas imobiliárias e a rapidez com que, tanto o proprietário antigo obtenha uma resposta rápida, quanto para um possível novo proprietário, fique mais fácil a pesquisa e a decisão de compra de um imóvel, uma vez que o mesmo pode pesquisar no devido meio criado.

2. METODOLOGIA

2.1 BIBLIOTECAS UTILIZADAS

Foi empregada a biblioteca Tkinter que é responsável pela interface gráfica do programa.

2.2 FUNÇÕES CRIADAS

A biblioteca Tkinter utiliza, a cada botão, uma nova função. Portanto, a cada nova ação foi criada uma nova função totalizando 8 funções no total, sendo elas:

- 1- proxima_casa;
- 2- procurar;
- 3- salvando;
- 4- catalogo;
- 5- instruções;
- 6- propostas;
- 7- aceitar;
- 8- negar.

2.3 FUNCIONAMENTO DO PROGRAMA

Nas figuras 1 e 2 importamos a biblioteca Tkinter que é responsável pela interface gráfica. Após isso, abrimos uma janela principal com seus títulos, criamos várias listas vazias que serão utilizadas no programa e todas as funções que usamos no código.

FIGURA 1 – TELA PRINCIPAL – PARTE 2

```
# importando a biblioteca tkinter
from tkinter import *

# Abrindo janela principal
janela = Tk()
janela.title("New Application")
janela.geometry("1450x700+0+0")

# Informações que serão printadas na tela
Titulo1 = Label(janela, text="Olá, seja bem-vindo a Imobiliária MZZ", font=("Times", 42, "bold"), fg="black").place(x=200, y=100)
Titulo2 = Label(janela, text="Escolha o que você deseja fazer hoje:", font=("Times", 35, "bold"), fg="gray40").place(x=275, y=200)

# Variáveis a serem utilizadas depois
casas, casa, aceitadas, filtros = [], [], [], []

# Chamada quando clicada o botão "próxima" da tela 4
def proxima_casa(janela4):...

# Chamada quando clicado o botão "procurar" da tela 2
def procurar(janela2, entry_bairro, entry_area, entry_quartos, entry_banheiro, entry_valor):...

# Chamada quando clicado o botão "enviar" da tela 1
# Tem por objetivo salvar todas as informacoes da casa que foram escritas
def salvando(janela1, entry_nome, entry_endereco, entry_bairro, entry_area, entry_quartos, entry_banheiro,
             entry_descrição, entry_email, entry_valor):...
```

FIGURA 2 – TELA PRINCIPAL – PARTE 2

```
# Chamada quando clicado o botão "vender" da tela principal
# Tem por objetivo fornecer um formulário para o usuário colocar os dados de sua casa que pretende vender
def catalogo():...

# Chamada pelo botão "comprar" da tela principal
def instrucoes():...

# Chamada quando clicado no botão "funcionario" da tela principal
def propostas():...

# Chamada quando clicado o botão "aceitar" da tela 3
def aceitar(janela3):...

# Chamada quando clicado o botão "negar" da tela 3
def negar(janela3):...
```


FIGURA 3 – TELA PRINCIPAL – PARTE 3

```
# Botões que aparecem na tela principal
# Abre a tela 1 chamando a função "catalogo"
Vender = Button(janela, text="Vender", width=18, height=2, bg="snow", command=catalogo).place(x=400, y=350)
# Abre a tela 2 chamando a função "instrucoes"
Comprar = Button(janela, text="Comprar", width=18, height=2, bg="snow", command=instrucoes).place(x=750, y=350)
# Abre a tela 3 chamando a função "propostas"
Funcionario = Button(janela, text="Você é funcionário?", width=18, height=2, bg="snow", command=propostas).place(x=1200, y=650)

janela.mainloop()
```

Criamos 3 botões na tela principal dos quais o 1º é apertado pelo usuário caso ele deseje vender uma casa, o 2º é clicado caso ele deseje comprar uma casa e 3º é utilizado pelo funcionário da imobiliária.

FIGURA 4 – FUNÇÃO CATALOGO – PARTE 1

```
def catalogo():
    # Cria uma nova janela
    janela1 = Toplevel(janela)
    janela1.geometry("1450x700+0+0")

    # Títulos da tela
    heading2 = Label(janela1, text="Primeiros passos", font=("Times", 42, "bold"), fg="black").place(x=450, y=100)
    heading3 = Label(janela1, text="Dados da casa:", font=("Times", 35, "bold"), fg="gray40").place(x=90, y=200)
```

A função catalogo abre uma nova janela com novos títulos.

Nas figuras 5, 6 e 7 pedimos ao usuário seus dados e os dados da casa que ele deseja vender.

FIGURA 5– FUNÇÃO CATALOGO – PARTE 2

```
# A seguir a variável "entry_N" é o espaço para se escrever o "N" do proprietário
# A seguir a variável "N" printa "N" na tela

# Exemplo
# A variável "entry_nome" é o espaço para quem deseja escrever
# A variável "Nome" printa "Nome:" na tela

entry_nome = Entry(janela1, width=100, bg="snow")
entry_nome.place(x=250, y=305)
Nome = Label(janela1, text="Nome completo:", font=("Times", 15), fg="black")
Nome.place(x=90, y=300)

entry_endereco = Entry(janela1, width=100, bg="snow")
entry_endereco.place(x=250, y=330)
Endereco = Label(janela1, text="Endereço:", font=("Times", 15), fg="black")
Endereco.place(x=90, y=325)
```

FIGURA 6– FUNÇÃO CATALOGO – PARTE 3

```

entry_bairro = Entry(janela1, width=100, bg="snow")
entry_bairro.place(x=250, y=355)
Bairro = Label(janela1, text="Bairro:", font=("Times", 15), fg="black")
Bairro.place(x=90, y=350)

entry_area = Entry(janela1, width=100, bg="snow")
entry_area.place(x=250, y=380)
Area = Label(janela1, text="Área (m²):", font=("Times", 15), fg="black")
Area.place(x=90, y=375)

entry_quartos = Entry(janela1, width=100, bg="snow")
entry_quartos.place(x=250, y=405)
Quartos = Label(janela1, text="Qtddde de quartos:", font=("Times", 15), fg="black")
Quartos.place(x=90, y=400)

entry_banheiro = Entry(janela1, width=97, bg="snow")
entry_banheiro.place(x=270, y=430)
Banheiro = Label(janela1, text="Qtddde de banheiros:", font=("Times", 15), fg="black")
Banheiro.place(x=90, y=425)

entry_descricao = Entry(janela1, width=100, bg="snow")
entry_descricao.place(x=250, y=455)
Descricao = Label(janela1, text="Descrição:", font=("Times", 15), fg="black")
Descricao.place(x=90, y=450)

```

FIGURA 7– FUNÇÃO CATALOGO – PARTE 4

```

entry_email = Entry(janela1, width=100, bg="snow")
entry_email.place(x=250, y=480)
Email = Label(janela1, text="Email:", font=("Times", 15), fg="black")
Email.place(x=90, y=475)

entry_valor = Entry(janela1, width=100, bg="snow")
entry_valor.place(x=250, y=505)
Valor = Label(janela1, text="Valor (R$):", font=("Times", 15), fg="black")
Valor.place(x=90, y=500)

# Botão que aparece na tela para ser pressionado para enviar todos os dados para a imobiliária
# Chama a função "salvando"
Enviar = Button(janela1, text="Enviar", width=18, height=2, bg="snow",
                command=lambda: salvando(janela1, entry_nome, entry_endereco, entry_bairro, entry_area,
                                         entry_quartos, entry_banheiro, entry_descricao, entry_email, entry_valor))
Enviar.place(x=380, y=580)

```

Ao final da figura, temos um botão chamado Enviar o qual deve ser utilizado pelo usuário depois que digitado todas as informações de sua casa.

FIGURA 8– FUNÇÃO SALVANDO – PARTE 1

```
def salvando(janela1, entry_nome, entry_endereço, entry_bairro, entry_area, entry_quartos, entry_banheiro,
            entry_descrição, entry_email, entry_valor):
    # Variáveis global que é modificada dentro da função
    global casa
    casa = []
    # A seguir será realizado o mesmo procedimento para todos os dados "N" dados fornecidos pelo usuário
    # O comando ".get()" pega todo o texto que foi digitado pelo usuário
    nome = entry_nome.get()
    # colocamos essa variável dentro da lista
    casa.append(nome)
    # A cada 2 linhas acontece a mesma coisa relatada acima
    endereço = entry_endereço.get()
    casa.append(endereço)
    bairro = entry_bairro.get()
    casa.append(bairro)
    area = entry_area.get()
    casa.append(area)
    quartos = entry_quartos.get()
    casa.append(quartos)
    banheiros = entry_banheiro.get()
    casa.append(banheiros)
    descrição = entry_descrição.get()
    casa.append(descrição)
    email = entry_email.get()
    casa.append(email)
```

A função Salvando guarda todas as descrições fornecidas pelo usuário em uma lista chamada casa.

FIGURA 9– FUNÇÃO SALVANDO – PARTE 2

```
valor = entry_valor.get()
casa.append(valor)
# Verifica se o usuário digitou em todos os campos
if (casa[0] != "") and (casa[1] != "") and (casa[2] != "") and (casa[3] != "") and (casa[4] != "") and (
    casa[5] != "") and (casa[6] != "") and (casa[7] != "") and (casa[8] != ""):
    casa[3] = int(casa[3])
    casa[4] = int(casa[4])
    casa[5] = int(casa[5])
    casa[8] = float(casa[8])

    # Colocamos a lista na matriz "casas" na qual cada linha terá a informação de uma casa
    casas.append(casa)

    # Printamos na tela avisando o usuário que sua proposta foi enviada
    analise = Label(janela1, text="Sua proposta foi enviada!", font=("Times", 25), fg="black").place(x=380, y=580)
    analise1 = Label(janela1, text="Aguarde a resposta da imobiliária!", font=("Times", 25), fg="black").place(x=380, y=650)
# Se o usuário não digitou em algum campo
else:
    analise2 = Label(janela1, text="Campo obrigatório não preenchido", font=("Times", 25), fg="black").place(x=380, y=650)
```

Verifica-se se o usuário digitou todas as informações necessárias. Se sim, printa-se “Sua proposta foi enviada”, se não, printa-se “Campo obrigatório não preenchido”.

FIGURA 10– FUNÇÃO PROPOSTAS – PARTE 1

```
def propostas():
    # Abrindo uma nova janela
    janela3 = Toplevel(janela)
    janela3.geometry("1450x700+0+0")

    # Verificando se há propostas a serem analisadas pela imobiliária
    if len(casas) > 0:
        # Printando as informações da primeira casa disponível para avaliação
        Titulo = Label(janela3, text="Propostas em análise:", font=("Times", 42, "bold"), fg="black").place(x=350, y=80)

        Casa = Label(janela3, text="Casa:", font=("Times", 25, "bold"), fg="black").place(x=100, y=160)

        nome = Label(janela3, text=casas[0][0], font=("Times", 18, ), fg="gray40").place(x=240, y=202)
        Nome = Label(janela3, text="Proprietário:", font=("Times", 18, "bold"), fg="black").place(x=100, y=200)

        endereço = Label(janela3, text=casas[0][1], font=("Times", 18, ), fg="gray40").place(x=240, y=232)
        Endereco = Label(janela3, text="Endereço:", font=("Times", 18, "bold"), fg="black").place(x=100, y=230)

        bairro = Label(janela3, text=casas[0][2], font=("Times", 18, ), fg="gray40").place(x=240, y=262)
        Bairro = Label(janela3, text="Bairro:", font=("Times", 18, "bold"), fg="black").place(x=100, y=260)

        area = Label(janela3, text=casas[0][3], font=("Times", 18, ), fg="gray40").place(x=240, y=292)
        Area = Label(janela3, text="Área:", font=("Times", 18, "bold"), fg="black").place(x=100, y=290)
```

A função Propostas abre uma nova janela e, se houver casas a serem avaliadas, mostra as informações da mesma.

FIGURA 11– FUNÇÃO PROPOSTAS – PARTE 2

```
quartos = Label(janela3, text=casas[0][4], font=("Times", 18, ), fg="gray40").place(x=240, y=322)
Quartos = Label(janela3, text="Quartos:", font=("Times", 18, "bold"), fg="black").place(x=100, y=320)

banheiros = Label(janela3, text=casas[0][5], font=("Times", 18, ), fg="gray40").place(x=240, y=352)
Banheiros = Label(janela3, text="Banheiros:", font=("Times", 18, "bold"), fg="black").place(x=100, y=350)

descricao = Label(janela3, text=casas[0][6], font=("Times", 18, ), fg="gray40").place(x=240, y=382)
Descricao = Label(janela3, text="Descrição:", font=("Times", 18, "bold"), fg="black").place(x=100, y=380)

valor = Label(janela3, text=casas[0][8], font=("Times", 18, ), fg="gray40").place(x=240, y=412)
Valor = Label(janela3, text="Valor:", font=("Times", 18, "bold"), fg="black").place(x=100, y=410)

# Botão "Aceitar" que deverá ser utilizado caso o funcionário queira aceitar a proposta do proprietário
# Tal botão chama a função "aceitar"
Aceitar = Button(janela3, text="Aceitar", width=18, height=2, bg="snow", command=lambda: aceitar(janela3)).place(x=200, y=550)

# Botão "Negar" que deverá ser utilizado caso o funcionário queira negar a proposta do proprietário
# Tal botão chama a função "negar"
Negar = Button(janela3, text="Negar", width=18, height=2, bg="snow", command=lambda: negar(janela3)).place(x=400, y=550)

# Se não houver casas disponíveis para avaliação é printado "Não há casas a avaliar"
else:
    Acabou = Label(janela3, text="Não há casas a avaliar.", font=("Times", 35, "bold"), fg="black").place(x=400, y=350)
```

Dada todas as informações da casa ao funcionário da imobiliária, ele terá duas opções: 1º-Aceitar a proposta clicando no botão Aceitar que chamará a função aceitar. 2º- Negar a proposta clicando no botão Negar que chamará a função negar.

FIGURA 12– FUNÇÃO ACEITAR– PARTE 1

```
def aceitar(janela3):
    global casas
    global aceitadas
    # A casa aceita é colocada na matriz que contém todas as casas aceitas
    aceitadas.append(casas[0])
    # Deleta tal casa da matriz casas para que haja uma nova casa na posição 0 da matriz a ser analisada pelo funcionário
    del (casas[0])
    # Fechando a tela 3
    janela3.destroy()

    # Abrindo uma nova janela
    janela3 = Toplevel(janela)
    janela3.geometry("1450x700+0+0")
    # Verificando se há propostas a serem analisadas pela imobiliária
    if len(casas) > 0:
        # Printando as informações da casa disponível para avaliação
        Titulo = Label(janela3, text="Propostas em análise:", font=("Times", 42, "bold"), fg="black").place(x=350, y=80)

        casa = Label(janela3, text="Casa:", font=("Times", 25, "bold"), fg="black").place(x=100, y=160)

        nome = Label(janela3, text=casas[0][0], font=("Times", 18, ), fg="gray40").place(x=240, y=202)
        Nome = Label(janela3, text="Proprietário:", font=("Times", 18, "bold"), fg="black").place(x=100, y=200)

        endereco = Label(janela3, text=casas[0][1], font=("Times", 18), fg="gray40").place(x=240, y=232)
        Endereco = Label(janela3, text="Endereço:", font=("Times", 18, "bold"), fg="black").place(x=100, y=230)

        bairro = Label(janela3, text=casas[0][2], font=("Times", 18), fg="gray40").place(x=240, y=262)
        Bairro = Label(janela3, text="Bairro:", font=("Times", 18, "bold"), fg="black").place(x=100, y=260)
```

Essa função coloca as casas aceitas pelo funcionário em uma matriz chamada aceitadas. Após isso ela deleta tal casa da matriz casas, fecha a janela do funcionário e abre uma próxima janela com uma próxima casa enquanto houver casas a serem avaliadas.

FIGURA 13– FUNÇÃO ACEITAR– PARTE 2

```
area = Label(janela3, text=casas[0][3], font=("Times", 18), fg="gray40").place(x=240, y=292)
Area = Label(janela3, text="Área:", font=("Times", 18, "bold"), fg="black").place(x=100, y=290)

quartos = Label(janela3, text=casas[0][4], font=("Times", 18), fg="gray40").place(x=240, y=322)
Quartos = Label(janela3, text="Quartos:", font=("Times", 18, "bold"), fg="black").place(x=100, y=320)

banheiros = Label(janela3, text=casas[0][5], font=("Times", 18), fg="gray40").place(x=240, y=352)
Banheiros = Label(janela3, text="Banheiros:", font=("Times", 18, "bold"), fg="black").place(x=100, y=350)

descricao = Label(janela3, text=casas[0][6], font=("Times", 18), fg="gray40").place(x=240, y=382)
Descricao = Label(janela3, text="Descrição:", font=("Times", 18, "bold"), fg="black").place(x=100, y=380)

valor = Label(janela3, text=casas[0][8], font=("Times", 18), fg="gray40").place(x=240, y=412)
Valor = Label(janela3, text="Valor:", font=("Times", 18, "bold"), fg="black").place(x=100, y=410)

# Botão "Aceitar" que deverá ser utilizado caso o funcionário queira aceitar a proposta do proprietário
# Tal botão chama a função "aceitar"

Aceitar = Button(janela3, text="Aceitar", width=18, height=2, bg="snow", command=lambda: aceitar(janela3)).place(x=200, y=550)

# Botão "Negar" que deverá ser utilizado caso o funcionário queira negar a proposta do proprietário
# Tal botão chama a função "negar"

Negar = Button(janela3, text="Negar", width=18, height=2, bg="snow", command=lambda: negar(janela3)).place(x=400, y=550)

# Se não houver casas disponíveis para avaliação é printado "Não há casas a avaliar"
else:
    Acabou = Label(janela3, text="Não há casas a avaliar.", font=("Times", 35, "bold"), fg="black").place(x=400, y=350)
```

O funcionário terá novamente o direito de aceitar ou negar a proposta e, quando respondido a todas as propostas será mostrado “Não há casas a avaliar”.

FIGURA 14– FUNÇÃO NEGAR– PARTE 1

```
def negar(janela3):
    global casas
    # Deleta tal casa da matriz casas para que haja uma nova casa na posição 0 da matriz a ser analisada pelo funcionário
    del (casas[0])
    # Fechando a tela 3
    janela3.destroy()
    # Abrindo uma nova janela
    janela3 = Toplevel(janela)
    janela3.geometry("1450x700+0+0")

    # Verificando se há propostas a serem analisadas pela imobiliária
    if len(casas) > 0:
        # Printando as informações da casa disponível para avaliação
        Titulo = Label(janela3, text="Propostas em análise:", font=("Times", 42, "bold"), fg="black").place(x=350, y=80)

        casa = Label(janela3, text="Casa:", font=("Times", 25, "bold"), fg="black").place(x=100, y=160)

        nome = Label(janela3, text=casas[0][0], font=("Times", 18, ), fg="gray40").place(x=240, y=202)
        Nome = Label(janela3, text="Proprietário:", font=("Times", 18, "bold"), fg="black").place(x=100, y=200)

        endereco = Label(janela3, text=casas[0][1], font=("Times", 18), fg="gray40").place(x=240, y=232)
        Endereco = Label(janela3, text="Endereço:", font=("Times", 18, "bold"), fg="black").place(x=100, y=230)

        bairro = Label(janela3, text=casas[0][2], font=("Times", 18), fg="gray40").place(x=240, y=262)
        Bairro = Label(janela3, text="Bairro:", font=("Times", 18, "bold"), fg="black").place(x=100, y=260)
```

A função deleta a casa negada da matriz casas, fecha a janela do funcionário e abre uma próxima janela com uma próxima casa enquanto houver casas a serem avaliadas.

FIGURA 15– FUNÇÃO NEGAR– PARTE 2

```
area = Label(janela3, text=casas[0][3], font=("Times", 18), fg="gray40").place(x=240, y=292)
Area = Label(janela3, text="Área:", font=("Times", 18, "bold"), fg="black").place(x=100, y=290)

quartos = Label(janela3, text=casas[0][4], font=("Times", 18), fg="gray40").place(x=240, y=322)
Quartos = Label(janela3, text="Quartos:", font=("Times", 18, "bold"), fg="black").place(x=100, y=320)

banheiros = Label(janela3, text=casas[0][5], font=("Times", 18), fg="gray40").place(x=240, y=352)
Banheiros = Label(janela3, text="Banheiros:", font=("Times", 18, "bold"), fg="black").place(x=100, y=350)

descricao = Label(janela3, text=casas[0][6], font=("Times", 18), fg="gray40").place(x=240, y=382)
Descricao = Label(janela3, text="Descrição:", font=("Times", 18, "bold"), fg="black").place(x=100, y=380)

valor = Label(janela3, text=casas[0][8], font=("Times", 18), fg="gray40").place(x=240, y=412)
Valor = Label(janela3, text="Valor:", font=("Times", 18, "bold"), fg="black").place(x=100, y=410)

# Botão "Aceitar" que deverá ser utilizado caso o funcionário queira aceitar a proposta do proprietário
# Tal botão chama a função "aceitar"
Aceitar = Button(janela3, text="Aceitar", width=18, height=2, bg="snow",
                 command=lambda: aceitar(janela3)).place(x=200, y=550)

# Botão "Negar" que deverá ser utilizado caso o funcionário queira negar a proposta do proprietário
# Tal botão chama a função "negar"
Negar = Button(janela3, text="Negar", width=18, height=2, bg="snow", command=lambda: negar(janela3)).place(x=400, y=550)

# Se não houver casas disponíveis para avaliação é printado "Não há casas a avaliar"
else:
    Acabou = Label(janela3, text="Não há casas a avaliar.", font=("Times", 35, "bold"), fg="black").place(x=400, y=350)
```

O funcionário terá novamente o direito de aceitar ou negar a proposta e, quando respondido a todas as propostas será printado “Não há casas a avaliar”.

FIGURA 16 – FUNÇÃO INSTRUÇÕES

```
def instrucoes():
    # Abrindo tela 2
    janela2 = Toplevel(janela)
    janela2.geometry("1450x700+0+0")
    # Printando na tela 2
    heading1 = Label(janela2, text="Casas disponíveis:", font=("Times", 42, "bold"), fg="black").place(x=200, y=100)
    heading2 = Label(janela2, text="Filtros:", font=("Times", 30, "bold"), fg="gray40").place(x=100, y=200)

    # A seguir o comprador terá a oportunidade de escolher algumas especificações que ele deseja que haja em sua casa
    # A variável "entry_N" é o espaço para se escrever o "N" do proprietário
    # A variável "N" printa "N" na tela

    # Exemplo

    # A variável "entry_bairro" é o espaço para quem deseja escrever
    # A variável "Bairro" printa "Bairro:" na tela

    entry_bairro = Entry(janela2, width=100, bg="snow")
    entry_bairro.place(x=250, y=305)
    Bairro = Label(janela2, text="Bairro:", font=("Times", 15), fg="black")
    Bairro.place(x=90, y=300)

    entry_area = Entry(janela2, width=100, bg="snow")
    entry_area.place(x=250, y=330)
    Area = Label(janela2, text="Área (m²):", font=("Times", 15), fg="black")
    Area.place(x=90, y=325)
```

A função abre uma nova janela e dá ao usuário a possibilidade de escolher o Bairro em que ele deseja comprar uma casa e a área mínima dessa casa.

FIGURA 17– FUNÇÃO PROCURAR – PARTE 1

```
def procurar(janela2, entry_bairro, entry_area, entry_quartos, entry_banheiro, entry_valor):
    global aceitadas
    global filtros
    a = 0

    # O comando ".get()" pega todo o texto que foi digitado pelo usuário
    bairro = entry_bairro.get()
    area = int(entry_area.get())
    banheiros = int(entry_banheiro.get())
    valor = float(entry_valor.get())
    quartos = int(entry_quartos.get())

    # Para cada casa (linha) da matriz "aceitadas" verificaremos se as especificações condizem com as desejadas pelo usuário
    for i in range(len(aceitadas)):
        if bairro == aceitadas[i][2]:
            if area <= aceitadas[i][3]:
                if quartos <= aceitadas[i][4]:
                    if banheiros <= aceitadas[i][5]:
                        if valor >= aceitadas[i][8]:
                            # Se todas as condições condizem, colocamos a casa em uma matriz "filtros" na qual só há casas com os filtros selecionados
                            filtros.append(aceitadas[i])
                        else:
                            a = a + 1
                    else:
                        a = a + 1
                else:
                    a = a + 1
            else:
                a = a + 1
        else:
            a = a + 1
```

A função guarda em variáveis tudo que o usuário digitou anteriormente e verifica se há casas na matriz aceitadas que condizem com as especificações digitadas pelo comprador. Caso haja, ela guarda essa casa na matriz filtros.

FIGURA 18– FUNÇÃO INSTRUCOES – PARTE 2

```
entry_quartos = Entry(janela2, width=100, bg="snow")
entry_quartos.place(x=250, y=355)
Quartos = Label(janela2, text="Qtde de quartos:", font=("Times", 15), fg="black")
Quartos.place(x=90, y=350)

entry_banheiro = Entry(janela2, width=97, bg="snow")
entry_banheiro.place(x=270, y=380)
Banheiro = Label(janela2, text="Qtde de banheiros:", font=("Times", 15), fg="black")
Banheiro.place(x=90, y=375)

entry_valor = Entry(janela2, width=100, bg="snow")
entry_valor.place(x=250, y=405)
Valor = Label(janela2, text="Valor (R$):", font=("Times", 15), fg="black")
Valor.place(x=90, y=400)

# Depois de colocar todas as especificações da casa que ele ambiciona, o comprador clicara no botão "enviar"
Procurar = Button(janela2, text="Procurar", width=18, height=2, bg="snow",
                  command=lambda: procurar(janela2, entry_bairro, entry_area, entry_quartos, entry_banheiro,
                                           entry_valor))
Procurar.place(x=350, y=550)
```

Essa função também dá o direito de escolher quantidades mínimas de quartos, banheiros e quantidade máxima do valor. Após isso, o usuário clica em no botão procurar que chamará a função procurar.

FIGURA 19 – FUNÇÃO PROCURAR – PARTE 1

```
if a == len(aceitadas):
    janela5 = Toplevel()
    janela5.geometry("1450x700+0+0")
    encontradas = Label(janela5, text="Não possuem casas com a sua descrição!", font=("Times", 20, "bold")).place(x=100, y=140)
# Para o caso de haverem casas na matriz filtros
else:
    # Abre uma nova janela
    janela4 = Toplevel(janela)
    janela4.geometry("1450x700+0+0")
    # Printa na tela a casa que condiz com o que deseja o comprador
    casa = Label(janela4, text="Casa:", font=("Times", 25, "bold"), fg="black").place(x=100, y=140)

    nome = Label(janela4, text=filtros[0][0], font=("Times", 18), fg="gray40").place(x=240, y=202)
    Nome = Label(janela4, text="Proprietário:", font=("Times", 18, "bold"), fg="black").place(x=100, y=200)

    endereco = Label(janela4, text=filtros[0][1], font=("Times", 18), fg="gray40").place(x=240, y=232)
    Endereco = Label(janela4, text="Endereço:", font=("Times", 18, "bold"), fg="black").place(x=100, y=230)

    bairro = Label(janela4, text=filtros[0][2], font=("Times", 18), fg="gray40").place(x=240, y=262)
    Bairro = Label(janela4, text="Bairro:", font=("Times", 18, "bold"), fg="black").place(x=100, y=260)

    area = Label(janela4, text=filtros[0][3], font=("Times", 18), fg="gray40").place(x=240, y=292)
    Area = Label(janela4, text="Área:", font=("Times", 18, "bold"), fg="black").place(x=100, y=290)

    quartos = Label(janela4, text=filtros[0][4], font=("Times", 18), fg="gray40").place(x=240, y=322)
    Quartos = Label(janela4, text="Quartos:", font=("Times", 18, "bold"), fg="black").place(x=100, y=320)
```


Caso não haja nenhuma casa com as especificações do usuário é printado “Não possuem casas com a sua descrição!”. Caso haja, printa-se as informações dessa casa.

FIGURA 20 – FUNÇÃO PROXIMA_CASA – PARTE 1

```
def proxima_casa(janela4):
    # Destruí a tela 4
    janela4.destroy()
    # Deleta a casa da posição 0 da matriz "filtros"
    del (filtros[0])
    # Abre uma nova tela 4
    janela4 = Toplevel(janela)
    janela4.geometry("1450x700+0+0")
    # Printa na tela a próxima casa que condiz com o que deseja o comprador
    casa = Label(janela4, text="Casa:", font=("Times", 25, "bold"), fg="black").place(x=100, y=140)

    nome = Label(janela4, text=filtros[0][0], font=("Times", 18, ), fg="gray40").place(x=240, y=202)
    Nome = Label(janela4, text="Proprietário:", font=("Times", 18, "bold"), fg="black").place(x=100, y=200)

    endereco = Label(janela4, text=filtros[0][1], font=("Times", 18, ), fg="gray40").place(x=240, y=232)
    Endereco = Label(janela4, text="Endereço:", font=("Times", 18, "bold"), fg="black").place(x=100, y=230)

    bairro = Label(janela4, text=filtros[0][2], font=("Times", 18, ), fg="gray40").place(x=240, y=262)
    Bairro = Label(janela4, text="Bairro:", font=("Times", 18, "bold"), fg="black").place(x=100, y=260)

    area = Label(janela4, text=filtros[0][3], font=("Times", 18, ), fg="gray40").place(x=240, y=292)
    Area = Label(janela4, text="Área:", font=("Times", 18, "bold"), fg="black").place(x=100, y=290)

    quartos = Label(janela4, text=filtros[0][4], font=("Times", 18, ), fg="gray40").place(x=240, y=322)
    Quartos = Label(janela4, text="Quartos:", font=("Times", 18, "bold"), fg="black").place(x=100, y=320)
```

A função destrói a tela antiga e printa em uma nova tela todas a informações da próxima casa.

FIGURA 21 – FUNÇÃO PROXIMA_CASA – PARTE 2

```
banheiros = Label(janela4, text=filtros[0][5], font=("Times", 18, ), fg="gray40").place(x=240, y=352)
Banheiros = Label(janela4, text="Banheiros:", font=("Times", 18, "bold"), fg="black").place(x=100, y=350)

descricao = Label(janela4, text=filtros[0][6], font=("Times", 18, ), fg="gray40").place(x=240, y=382)
Descricao = Label(janela4, text="Descrição:", font=("Times", 18, "bold"), fg="black").place(x=100, y=380)

valor = Label(janela4, text=filtros[0][8], font=("Times", 18, ), fg="gray40").place(x=240, y=412)
Valor = Label(janela4, text="Valor:", font=("Times", 18, "bold"), fg="black").place(x=100, y=410)

contato = Label(janela4, text="Para mais informações entre em contato conosco:", font=("Times", 10, "bold"), fg="black").place(x=900, y=550)
telefone = Label(janela4, text="Telefong: (41)99948-2502", font=("Times", 10, "bold"), fg="black").place(x=915, y=575)
email = Label(janela4, text="Email: imobiliariamz@gmail.com", font=("Times", 10, "bold"), fg="black").place(x=910, y=600)

# Se ainda houver mais de uma casa na matriz filtros, o usuário ainda pode clicar no botão "proximo" para aparecer a próxima casa
if len(filtros) > 1:
    proximo = Button(janela4, text="Próximo", width=10, height=2, bg="snow",
                     command=lambda: proxima_casa(janela4)).place(x=450, y=500)
# Se não houver mais de uma casa, printamos "Não há mais casas disponíveis"
else:
    mensagem = Label(janela4, text="Não há mais casas disponíveis!", font=("Times", 35), fg="black").place(x=150, y=500)
```

Além de printar as informações da casa, é printado as informações de contato com a imobiliária. Além disso, caso haja mais uma casa que condiz com as especificações do comprador, é disponibilizado um botão chamado próximo que chama a função próxima_casa, caso não haja, printa-se “Não há mais casas disponíveis”.

FIGURA 22 – FUNÇÃO PROCURAR – PARTE 3

```
banheiros = Label(janela4, text=filtros[0][5], font=("Times", 18), fg="gray40").place(x=240, y=352)
Banheiros = Label(janela4, text="Banheiros:", font=("Times", 18, "bold"), fg="black").place(x=100, y=350)

descricao = Label(janela4, text=filtros[0][6], font=("Times", 18), fg="gray40").place(x=240, y=382)
Descrição = Label(janela4, text="Descrição:", font=("Times", 18, "bold"), fg="black").place(x=100, y=380)

valor = Label(janela4, text=filtros[0][8], font=("Times", 18), fg="gray40").place(x=240, y=412)
Valor = Label(janela4, text="Valor:", font=("Times", 18, "bold"), fg="black").place(x=100, y=410)

contato = Label(janela4, text="Para mais informações entre em contato conosco:", font=("Times", 10, "bold"), fg="black").place(x=900, y=550)
telefone = Label(janela4, text="Telefone: (41)99948-2502", font=("Times", 10, "bold"), fg="black").place(x=915, y=575)
email = Label(janela4, text="Email: imobiliariamrz@gmail.com", font=("Times", 10, "bold"), fg="black").place(x=910, y=600)

# Se haver mais de uma casa em filtros, para saber a próxima o usuário aperta o botão "proximo"
if len(filtros) > 1:
    proximo = Button(janela4, text="Próxima", width=18, height=2, bg="snow",
                     command=lambda: proxima_casa(janela4)).place(x=450, y=500)
# Se não houver mais de uma casa, printamos "Não há mais casas disponíveis"
else:
    mensagem = Label(janela4, text="Não há mais casas disponíveis!", font=("Times", 35), fg="black").place(x=150, y=500)
```

Além de printar as informações da casa, é printado as informações de contato com a imobiliária. Além disso, caso haja mais que uma casa que condiz com as especificações do comprador, é disponibilizado um botão chamado próximo que chama a função próxima_casa, caso não haja, printa-se “Não há mais casas disponíveis”.

2.4 RESULTADOS ENCONTRADOS

Após o usuário informar as características de sua casa e o perito imobiliário avaliar a mesma, aceitando-a, o imóvel é disponibilizado para a venda de acordo com a preferência do comprador.

2.5 DIFICULDADES ENCONTRADAS

Trabalhar com objetos não estudados em sala, como algumas bibliotecas e certas funcionalidades, por exemplo os botões, abrir e fechar janelas, salvar informações e mostrá-las em outros lugares.

2.6 SUGESTÕES PARA TRABALHOS POSTERIORES

- ♣ Snagging Parking Spaces na Universidade Federal do Paraná;
- ♣ Lista de presença online para os professores da Universidade Federal do Paraná;
- ♣ Cubo de Led que desenhe imagens;
- ♣ Sistema de irrigação que detecta a umidade no solo e rega as plantas quando necessário.

3. REFERÊNCIAS

- YouTube: Curso de Python- Módulo Tkinter- eXcript.

Disponível em:

<https://www.youtube.com/watch?v=kSQDp20eeqE&list=PLesCEcYj003ShHnUT83gQEH6KtG8uysUE>

- Explicação do Tkinter.

Disponível em:

<https://www.devmedia.com.br/tkinter-interfaces-graficas-em-python/33956>

- Explicação de Widget.

Disponível em:

<https://effbot.org/tkinterbook/entry.htm>