|  |
| --- |
| Aluno: Marcos Rodrigo Rodio C.P.F.: 226.408.218-65 |
| Disciplina: Interface Gráfica com o usuário em Python |
| Curso: Pós-Graduação em Desenvolvimento de Sistemas com Python |

**Orientações gerais**

1. Todos os campos de identificação acima deverão ser devidamente preenchidos;

2. O aluno deverá, obrigatoriamente, utilizar este formulário para enviar o MAPA. O sistema permite o envio de apenas UM arquivo, por isso, é necessário que você envie todas as informações de seu projeto neste documento e se certifique de estar enviando o arquivo correto;

3. Esta é uma atividade INDIVIDUAL, em caso de plágio, será zerada;

4. Procure ARGUMENTAR de forma clara e objetiva. Ao utilizar quaisquer materiais de pesquisa, cite e referencie conforme as normas da ABNT;

5. Formatação exigida no documento Word: Margens esquerda e superior de 3 cm e margens direita e inferior de 2 cm, fonte Arial, tamanho 12, espaçamento entre linhas 1,5 e alinhamento justificado, recuo no início de cada parágrafo de 1,25 cm.

6. Para realizar a Atividade MAPA, você precisa:

a. LER atentamente e CONSIDERAR o texto de apoio;

b. PESQUISAR sobre os TEMAS propostos;

c. Realizar e ENVIAR a atividade neste FORMULÁRIO PADRÃO

**ATIVIDADE MAPA SISTEMA DE LOCAÇÃO DE IMÓVEIS “PY ASSESSORIA”**

A IDE utilizada para a escrita dos código foi o VS Code e para construção das telas o QT Designer.

Como solicitado o Tanto o id de pessoa como do Imóvel é auto incremental.

Foi levado em conta a falta de familiaridade do Sr Roberto com a informática na escolha de um Layout simplista com poucas opções que poderiam dificultar sua utilização e sem muitas cores.

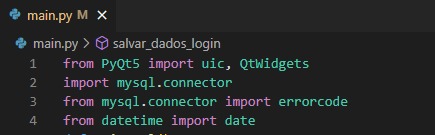
Os 03 requisitos (A interface gráfica da aplicação, A tabela do banco que irá receber os dados,A programação para armazenar os dados) foram atendidos como pode ser ver nos prints abaixo

Para se atender os requisitos de seleção foram utilizadas combo-box, também foram criadas abas no cadastro de imóvel para as descrições, observações e características.

Dividirei a apresentação dos prints em três partes e ao final deixarei o código completo e o link para acesso no github

1. **Código**

* **Imports.**

****

* **Funções.**

**Padrão do plano de fundo

Descrição gerada automaticamente**

**Texto

Descrição gerada automaticamente**

**Texto

Descrição gerada automaticamente**

**Texto

Descrição gerada automaticamente**

**Texto

Descrição gerada automaticamente**

**Texto

Descrição gerada automaticamente**

* **Main.**

**Texto

Descrição gerada automaticamente**

1. **Telas Em tempo de execução**

* **Login**

**Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente**

* **Inicial**

**Interface gráfica do usuário

Descrição gerada automaticamente com confiança média**

* **Cadastros de Pessoa e Imóvel**

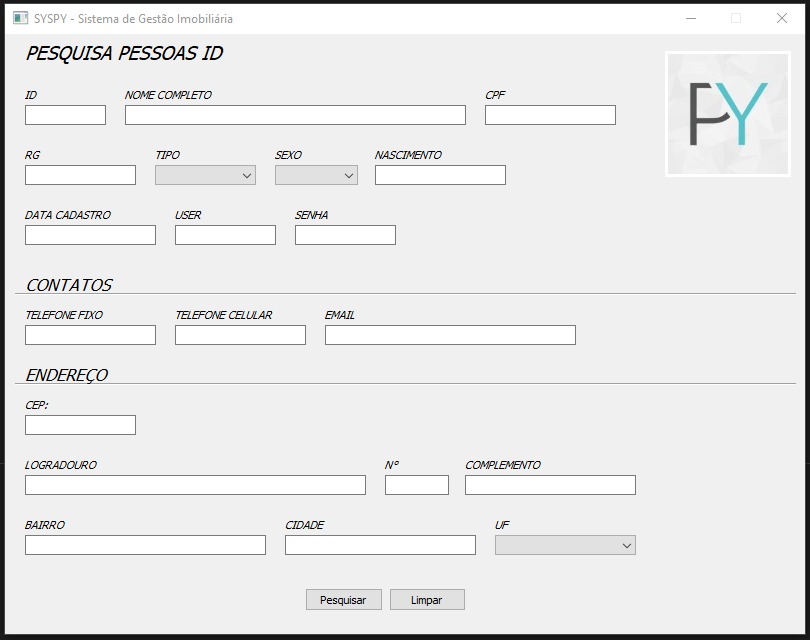
**Interface gráfica do usuário

Descrição gerada automaticamente**

**Interface gráfica do usuário

Descrição gerada automaticamente**

* **Pesquisa Pessoa e Imóvel via ID**

****

**Interface gráfica do usuário

Descrição gerada automaticamente**

* **QT Designer Telas Criadas**

**Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**

**Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente**

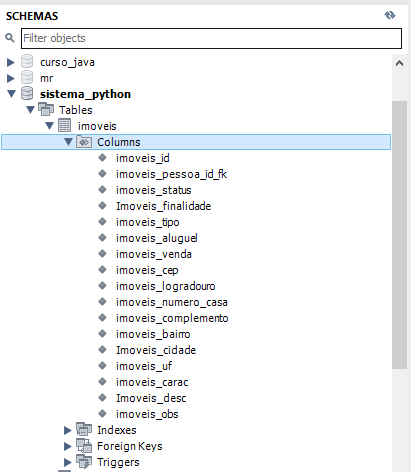
**Interface gráfica do usuário, Aplicativo, Tabela, Word

Descrição gerada automaticamente**

**Uma imagem contendo Gráfico

Descrição gerada automaticamente**

1. **Banco de Dados MY SQL**

**Interface gráfica do usuário

Descrição gerada automaticamente com confiança médiaInterface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente**

**Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente**

**Código Completo**

from PyQt5 import uic, QtWidgets

import mysql.connector

from mysql.connector import errorcode

from datetime import date

def c\_imovel():

    cadastroImovel.show()

def c\_pessoa():

    cadastroPessoa.show()

def p\_imovel():

    pesquisaImovel.show()

def p\_pessoa():

    pesquisaPessoa.show()

def cadastrar\_pessoa():

    try:

        pessoa\_nome = cadastroPessoa.pessoa\_nome\_le.text()

        pessoa\_sexo = cadastroPessoa.pessoa\_sexo\_cb.currentText()

        pessoa\_cpf = cadastroPessoa.pessoa\_cpf\_le.text()

        pessoa\_rg = cadastroPessoa.pessoa\_rg\_le.text()

        pessoa\_tipo = cadastroPessoa.pessoa\_tipo\_cb.currentText()

        pessoa\_user = cadastroPessoa.pessoa\_user\_le.text()

        pessoa\_senha = cadastroPessoa.pessoa\_senha\_le.text()

        pessoa\_data\_nascimento = cadastroPessoa.pessoa\_nasc\_de.date().toPyDate()

        pessoa\_tel\_fixo = cadastroPessoa.pessoa\_tel\_fixo\_le.text()

        pessoa\_celular = cadastroPessoa.pessoa\_celular\_le.text()

        pessoa\_email = cadastroPessoa.pessoa\_email\_le.text()

        pessoa\_cep = cadastroPessoa.pessoa\_cep\_le.text()

        pessoa\_logradouro = cadastroPessoa.pessoa\_logradouro\_le.text()

        pessoa\_numero\_casa = cadastroPessoa.pessoa\_numero\_casa\_le.text()

        pessoa\_complemento = cadastroPessoa.pessoa\_complemento\_le.text()

        pessoa\_bairro = cadastroPessoa.pessoa\_bairro\_le.text()

        pessoa\_cidade = cadastroPessoa.pessoa\_cidade\_le.text()

        pessoa\_uf = cadastroPessoa.pessoa\_uf\_cb.currentText()

        db\_connection = mysql.connector.connect(host='localhost', user='root', password='Marcos@1020', database='sistema\_python')

        cursor = db\_connection.cursor()

        sql = f"""INSERT INTO PESSOAS(PESSOA\_NOME, PESSOA\_SEXO, PESSOA\_CPF, PESSOA\_RG, PESSOA\_TIPO,PESSOA\_USER, PESSOA\_SENHA,

        PESSOA\_DATA\_NASCIMENTO,PESSOA\_DATA\_CADASTRO, PESSOA\_FONE\_FIXO, PESSOA\_CELULAR, PESSOA\_EMAIL,PESSOA\_CEP, PESSOA\_LOGRADOURO,

        PESSOA\_NUMERO\_CASA, PESSOA\_COMPLEMENTO,PESSOA\_BAIRRO, PESSOA\_CIDADE, PESSOA\_UF) VALUES ('{pessoa\_nome}','{pessoa\_sexo}' ,'{pessoa\_cpf}',

        '{pessoa\_rg}', '{pessoa\_tipo}','{pessoa\_user}', '{pessoa\_senha}','{pessoa\_data\_nascimento}', '{date.today()}', '{pessoa\_tel\_fixo}',

        '{pessoa\_celular}','{pessoa\_email}','{pessoa\_cep}','{pessoa\_logradouro}',{pessoa\_numero\_casa},'{pessoa\_complemento}','{pessoa\_bairro}',

        '{pessoa\_cidade}','{pessoa\_uf}')"""

        cursor.execute(sql)

        db\_connection.commit()

        db\_connection.close()

    except errorcode as e:

        print("Error while connecting to MySQL", e)

def cadastrar\_imovel():

    try:

        imoveis\_id = cadastroImovel.imoveis\_id\_pessoa\_le.text()

        imoveis\_status = cadastroImovel.imoveis\_satus\_cb.currentText()

        imoveis\_finalidade = cadastroImovel.imoveis\_finalidade\_cb.currentText()

        imoveis\_tipo = cadastroImovel.imoveis\_tipo\_cb.currentText()

        imoveis\_aluguel = cadastroImovel.imoveis\_aluguel\_le.text()

        imoveis\_venda = cadastroImovel.imoveis\_venda\_le.text()

        imoveis\_cep = cadastroImovel.Imoveis\_cep\_le.text()

        imoveis\_end = cadastroImovel.imoveis\_logradouro\_le.text()

        imoveis\_numero = cadastroImovel.imoveis\_numero\_le.text()

        imoveis\_complemento = cadastroImovel.imoveis\_complemento\_le.text()

        imoveis\_bairro = cadastroImovel.imoveis\_bairro\_le.text()

        imoveis\_cidade = cadastroImovel.imoveis\_cidade\_le.text()

        imoveis\_uf = cadastroImovel.imoveis\_estado\_cb.currentText()

        imoveis\_obs = cadastroImovel.imoveis\_obs\_te.toPlainText()

        imoveis\_carac = cadastroImovel.imoveis\_carac\_te.toPlainText()

        imoveis\_desc = cadastroImovel.imoveis\_desc\_te.toPlainText()

        db\_connection = mysql.connector.connect(host='localhost', user='root', password='Marcos@1020', database='sistema\_python')

        cursor = db\_connection.cursor()

        sql = f"""INSERT INTO imoveis (imoveis\_pessoa\_id\_fk, imoveis\_status, Imoveis\_finalidade, imoveis\_tipo, imoveis\_aluguel, imoveis\_venda,

        imoveis\_cep,imoveis\_logradouro,imoveis\_numero\_casa, imoveis\_complemento,imoveis\_bairro,Imoveis\_cidade,imoveis\_uf,imoveis\_carac,Imoveis\_desc,

        imoveis\_obs)

        values ({imoveis\_id},'{imoveis\_status}','{imoveis\_finalidade}','{imoveis\_tipo}',{imoveis\_aluguel},{imoveis\_venda}, '{imoveis\_cep}',

        '{imoveis\_end}',{imoveis\_numero},'{imoveis\_complemento}','{imoveis\_bairro}', '{imoveis\_cidade}','{imoveis\_uf}', '{imoveis\_obs}',

        '{imoveis\_carac}', '{imoveis\_desc}' )"""

        cursor.execute(sql)

        db\_connection.commit()

        db\_connection.close()

    except errorcode as e:

        print("Error while connecting to MySQL", e)

def pesquisar\_imoveis():

    try:

        db\_connection = mysql.connector.connect(host='localhost', user='root', password='Marcos@1020', database='sistema\_python')

        cursor = db\_connection.cursor(dictionary=True)

        pesquisa = pesquisaImovel.imoveis\_id\_imovel\_le.text()

        sql = f"select \* from imoveis Where imoveis\_id = '{pesquisa}'"

        cursor.execute(sql)

        resultado = cursor.fetchall()

        for rows in resultado:

            pesquisaImovel.imoveis\_id\_imovel\_le.setText(f"{rows['imoveis\_id']}")

            pesquisaImovel.imoveis\_id\_pessoa\_le.setText(f"{rows['imoveis\_pessoa\_id\_fk']}")

            pesquisaImovel.imoveis\_finalidade\_cb.setCurrentText(f"{rows['Imoveis\_finalidade']}")

            pesquisaImovel.imoveis\_satus\_cb.setCurrentText(f"{rows['imoveis\_status']}")

            pesquisaImovel.imoveis\_tipo\_cb.setCurrentText(f"{rows['imoveis\_tipo']}")

            pesquisaImovel.imoveis\_aluguel\_le.setText(f"{rows['imoveis\_aluguel']}")

            pesquisaImovel.imoveis\_venda\_le.setText(f"{rows['imoveis\_venda']}")

            pesquisaImovel.Imoveis\_cep\_le.setText(f"{rows['imoveis\_cep']}")

            pesquisaImovel.imoveis\_logradouro\_le.setText(f"{rows['imoveis\_logradouro']}")

            pesquisaImovel.imoveis\_numero\_le.setText(f"{rows['imoveis\_numero\_casa']}")

            pesquisaImovel.imoveis\_complemento\_le.setText(f"{rows['imoveis\_complemento']}")

            pesquisaImovel.imoveis\_bairro\_le.setText(f"{rows['imoveis\_bairro']}")

            pesquisaImovel.imoveis\_cidade\_le.setText(f"{rows['Imoveis\_cidade']}")

            pesquisaImovel.imoveis\_estado\_cb.setCurrentText(f"{rows['imoveis\_uf']}")

            pesquisaImovel.imoveis\_obs\_te.setPlainText(f"{rows['imoveis\_obs']}")

            pesquisaImovel.imoveis\_carac\_te.setPlainText(f"{rows['imoveis\_carac']}")

            pesquisaImovel.imoveis\_desc\_te.setPlainText(f"{rows['Imoveis\_desc']}")

        cursor.close()

        db\_connection.close()

    except errorcode as e:

        print("Error while connecting to MySQL", e)

def pesquisar\_pessoas():

    try:

        db\_connection = mysql.connector.connect(host='localhost', user='root', password='Marcos@1020', database='sistema\_python')

        cursor = db\_connection.cursor(dictionary=True)

        pesquisa = pesquisaPessoa.pessoa\_id\_le.text()

        cursor.execute(f"SELECT \* FROM pessoas WHERE pessoa\_id = {pesquisa} ")

        resultado = cursor.fetchall()

        for rows in resultado:

            pesquisaPessoa.nome\_le.setText(f"{rows['pessoa\_nome']}")

            pesquisaPessoa.cpf\_le.setText(f"{rows['pessoa\_cpf']}")

            pesquisaPessoa.rg\_le.setText(f"{rows['pessoa\_rg']}")

            pesquisaPessoa.tipo\_cb.setCurrentText(f"{rows['pessoa\_tipo']}")

            pesquisaPessoa.sexo\_cb.setCurrentText(f"{rows['pessoa\_sexo']}")

            pesquisaPessoa.nascimento\_le.setText(f"{rows['pessoa\_data\_nascimento']}")

            pesquisaPessoa.cadastro\_le.setText(f"{rows['pessoa\_data\_cadastro']}")

            pesquisaPessoa.user\_le.setText(f"{rows['pessoa\_user']}")

            pesquisaPessoa.senha\_le.setText(f"{rows['pessoa\_senha']}")

            pesquisaPessoa.fixo\_le.setText(f"{rows['pessoa\_fone\_fixo']}")

            pesquisaPessoa.celular\_le.setText(f"{rows['pessoa\_celular']}")

            pesquisaPessoa.email\_le.setText(f"{rows['pessoa\_email']}")

            pesquisaPessoa.cep\_le.setText(f"{rows['pessoa\_cep']}")

            pesquisaPessoa.endereco\_le.setText(f"{rows['pessoa\_logradouro']}")

            pesquisaPessoa.numero\_le.setText(f"{rows['pessoa\_numero\_casa']}")

            pesquisaPessoa.complemento\_le.setText(f"{rows['pessoa\_complemento']}")

            pesquisaPessoa.bairro\_le.setText(f"{rows['pessoa\_bairro']}")

            pesquisaPessoa.cidade\_le.setText(f"{rows['pessoa\_cidade']}")

            pesquisaPessoa.estado\_cb.setCurrentText(f"{rows['pessoa\_uf']}")

        cursor.close()

        db\_connection.close()

    except errorcode as e:

        print("Error while connecting to MySQL", e)

def limpar\_pesquisa\_pessoa():

   pesquisaPessoa.pessoa\_id\_le.setText("0")

   pesquisaPessoa.nome\_le.setText("")

   pesquisaPessoa.cpf\_le.setText("")

   pesquisaPessoa.rg\_le.setText("")

   pesquisaPessoa.tipo\_cb.setCurrentText("")

   pesquisaPessoa.sexo\_cb.setCurrentText("")

   pesquisaPessoa.nascimento\_le.setText("")

   pesquisaPessoa.cadastro\_le.setText("")

   pesquisaPessoa.user\_le.setText("")

   pesquisaPessoa.senha\_le.setText("")

   pesquisaPessoa.fixo\_le.setText("")

   pesquisaPessoa.celular\_le.setText("")

   pesquisaPessoa.email\_le.setText("")

   pesquisaPessoa.cep\_le.setText("")

   pesquisaPessoa.endereco\_le.setText("")

   pesquisaPessoa.numero\_le.setText("")

   pesquisaPessoa.complemento\_le.setText("")

   pesquisaPessoa.bairro\_le.setText("")

   pesquisaPessoa.cidade\_le.setText("")

   pesquisaPessoa.estado\_cb.setCurrentText("")

def limpar\_pesquisa\_imovel():

    pesquisaImovel.imoveis\_id\_imovel\_le.setText("0")

    pesquisaImovel.imoveis\_id\_pessoa\_le.setText("")

    pesquisaImovel.imoveis\_finalidade\_cb.setCurrentText("")

    pesquisaImovel.imoveis\_satus\_cb.setCurrentText("")

    pesquisaImovel.imoveis\_tipo\_cb.setCurrentText("")

    pesquisaImovel.imoveis\_aluguel\_le.setText("")

    pesquisaImovel.imoveis\_venda\_le.setText("")

    pesquisaImovel.Imoveis\_cep\_le.setText("")

    pesquisaImovel.imoveis\_logradouro\_le.setText("")

    pesquisaImovel.imoveis\_numero\_le.setText("")

    pesquisaImovel.imoveis\_complemento\_le.setText("")

    pesquisaImovel.imoveis\_bairro\_le.setText("")

    pesquisaImovel.imoveis\_cidade\_le.setText("")

    pesquisaImovel.imoveis\_estado\_cb.setCurrentText("")

    pesquisaImovel.imoveis\_obs\_te.setPlainText("")

    pesquisaImovel.imoveis\_carac\_te.setPlainText("")

    pesquisaImovel.imoveis\_desc\_te.setPlainText("")

def salvar\_dados\_login():

    try:

        db\_connection = mysql.connector.connect(host='localhost', user='root', password='Marcos@1020', database='sistema\_python')

        user = login.Login\_Edit.text()

        senha = login.Senha\_Edit.text()

        cursor = db\_connection.cursor(dictionary=True)

        cursor.execute(f"""SELECT pessoa\_senha FROM pessoas WHERE pessoa\_user = '{user}'""")

        resultado = cursor.fetchone()

        db\_connection.close()

        if(resultado == None):

            login.Login\_Edit.setText("")

            login.Senha\_Edit.setText("")

            login.label.setText("Usuário ou senha inválidos!")

            login.label.setStyleSheet("color:red")

        elif senha != resultado['pessoa\_senha']:

            login.Login\_Edit.setText("")

            login.Senha\_Edit.setText("")

            login.label.setText("Usuário ou senha inválidos!")

            login.label.setStyleSheet("color:red")

        else:

          Inicial.show()

          login.destroy()

    except errorcode as e:

        print("Error while connecting to MySQL", e)

def limpar\_login():

    login.Login\_Edit.setText("")

    login.Senha\_Edit.setText("")

    login.label.setText("")

app=QtWidgets.QApplication([])

login = uic.loadUi("login.ui")

Inicial = uic.loadUi("Inicial.ui")

cadastroPessoa = uic.loadUi("CadastroPessoas.ui")

cadastroImovel = uic.loadUi("CadastroImoveis.ui")

pesquisaImovel = uic.loadUi("PesquisaImoveis.ui")

pesquisaPessoa = uic.loadUi("Pesquisa.ui")

login.Button\_Cadastrar.clicked.connect(salvar\_dados\_login)

login.Button\_Cancelar.clicked.connect(limpar\_login)

Inicial.CadastrarImovel.clicked.connect(c\_imovel)

Inicial.CadastrarPessoa.clicked.connect(c\_pessoa)

Inicial.PesquisarImovel.clicked.connect(p\_imovel)

Inicial.PesquisarPessoa.clicked.connect(p\_pessoa)

cadastroPessoa.pessoa\_button\_incluir.clicked.connect(cadastrar\_pessoa)

cadastroImovel.pushButton\_incluir.clicked.connect(cadastrar\_imovel)

pesquisaPessoa.pushButton\_pesquisar.clicked.connect(pesquisar\_pessoas)

pesquisaImovel.pushButton\_pesquisar.clicked.connect(pesquisar\_imoveis)

pesquisaPessoa.pushButton\_limpar.clicked.connect(limpar\_pesquisa\_pessoa)

pesquisaImovel.pushButton\_limpar.clicked.connect(limpar\_pesquisa\_imovel)

login.show()

app.exec()

**REFERÊNCIAS**

Documentação QT Designer

Disponível em <https://doc.qt.io/qtforpython/overviews/stylesheet-syntax.html>

Acessado em 08 Dez 2022

Git [MarcosRodio/P-s\_Python (github.com)](https://github.com/MarcosRodio/P-s_Python)

****