

## Criando uma Agenda Telefônica

**Objetivo:** Desenvolver uma agenda telefônica em Python para praticar conceitos de programação como classes, métodos, listas e interações com o usuário. Este guia passo a passo ajudará a criar o projeto de forma clara e detalhada.

### Introdução

Neste projeto, você irá criar uma agenda telefônica que permite adicionar, remover, pesquisar e listar contatos. A agenda será estruturada usando a linguagem de programação Python.

### Passo 1: Inicializando a Classe da Agenda Telefônica

#### Criar a Classe:

Comece definindo uma classe chamada `AgendaTelefonica` para representar a agenda.

Dentro da classe, crie um método `__init__` que inicializa a agenda com um dicionário vazio para armazenar os contatos.

#### Explicação:

A classe `AgendaTelefonica` conterá um dicionário `self.contatos` onde as chaves serão os nomes dos contatos e os valores serão os números de telefone.

### Passo 2: Adicionar Contatos

#### Criar Método para Adicionar Contato:

Dentro da classe `AgendaTelefonica`, crie um método chamado `adicionar_contato`.

Este método deve receber dois parâmetros: nome e telefone.

Verifique se o contato já existe. Se existir, imprima uma mensagem informando que o contato já existe. Caso contrário, adicione o contato ao dicionário e imprima uma mensagem de sucesso.

#### Explicação:

Este método permitirá ao usuário adicionar novos contatos à agenda. Se o contato já existir, uma mensagem de aviso será exibida. Caso contrário, o contato será adicionado com sucesso.





### Passo 3: Remover Contatos

Criar Método para Remover Contato:

Dentro da classe AgendaTelefonica, crie um método chamado `remover_contato`.

Este método deve receber um parâmetro: `nome`.

Verifique se o contato existe. Se existir, remova-o do dicionário e imprima uma mensagem de sucesso. Caso contrário, imprima uma mensagem informando que o contato não foi encontrado.

Explicação:

Este método permitirá ao usuário remover contatos existentes da agenda. Se o contato não for encontrado, uma mensagem de aviso será exibida.

### Passo 4: Pesquisar Contatos

Criar Método para Pesquisar Contato:

Dentro da classe AgendaTelefonica, crie um método chamado `pesquisar_contato`.

Este método deve receber um parâmetro: `nome`.

Verifique se o contato existe no dicionário. Se existir, imprima o nome e o telefone do contato. Caso contrário, imprima uma mensagem informando que o contato não foi encontrado.

Explicação:

Este método permitirá ao usuário pesquisar por contatos específicos na agenda. Se o contato não for encontrado, uma mensagem de aviso será exibida.

### Passo 5: Listar Contatos

Criar Método para Listar Contatos:

Dentro da classe AgendaTelefonica, crie um método chamado `listar_contatos`.

Verifique se o dicionário de contatos está vazio. Se estiver, imprima uma mensagem informando que a agenda está vazia. Caso contrário, liste todos os contatos com seus respectivos telefones.

Explicação:

Este método permitirá ao usuário ver todos os contatos armazenados na agenda. Se não houver contatos, uma mensagem de aviso será exibida.

#### Passo 6: Criar o Menu de Opções

Criar Função para o Menu:

Fora da classe AgendaTelefonica, crie uma função chamada menu.

Dentro desta função, crie um loop que exiba as opções do menu e permita ao usuário escolher uma opção.

As opções do menu devem incluir adicionar, remover, pesquisar e listar contatos, além de uma opção para sair do programa.

Explicação:

Esta função será responsável por interagir com o usuário e chamar os métodos apropriados da classe AgendaTelefonica com base na escolha do usuário.

#### Passo 7: Finalizar o Programa

Adicionar Condição para Encerrar o Programa:

No menu, inclua uma opção para sair do programa.

Se o usuário escolher esta opção, imprima uma mensagem de despedida e encerre o loop.

Explicação:

Esta parte permitirá ao usuário encerrar o programa de maneira amigável, exibindo uma mensagem de agradecimento antes de sair.

