

അമ്മമാ  
മാ  
അമ്മമാ

# Descrição:

Desenvolva um programa de agenda de contatos que contenha todas as funcionalidades de uma agenda. Exemplo: Nome, telefone, e-mail e que permita ao usuário adicionar, excluir contatos. Quando o aplicativo for aberto, exibir todos os contatos cadastrados em ordem alfabética. E sempre que um usuário for cadastrado, enviar uma mensagem, dando as boas-vindas.

# *Bibliotecas usadas:*

```
import pywhatkit as kit  
import datetime  
from re import fullmatch
```

# Main:

```
6
7   def main():
8       |   menu()
9
10  main()
```

# Menu:

```
def menu():
    opcao = input('
.....
| | | | | Agenda de Contatos
Menu
[1] Cadastrar Contato.
[2] Listar Contato.
[3] Deletar Contato.
[4] Buscar Contato.
[5] Sair.
.....
Escolha uma opção acima: ')
    if opcao == "1":
        cadastrarContato()
    elif opcao == "2":
        listarContato()
    elif opcao == "3":
        deletarContato()
    elif opcao == '4':
        buscarContato()
    elif opcao == '5':
        sair()
    else:
        print ("Erro!")
```

# Cadastrar Contato:

```
def cadastrarContato():
    cadastrado=False
    identificacao = input("Escolha a identificação do contato: ")
    nome = input ("Informe o nome do contato: ")
    while True:
        telefone = input ("Informe o número de telefone do contato: ")
        if not fullmatch(r"^\+?[0-9]{2,4}\s?[0-9]{9,15}", telefone):
            print ("\n\33[1;31;40mNúmero de telefone incorreto.\33[m\n")
        else:
            break

    while True:
        email = input ("Informe o e-mail do contato: ")
        if checar_email(email) == True:
            break

    try:
        agenda = open("agenda.txt", "a")
        dados = f'{identificacao.upper()}; {nome.upper()}; {telefone}; {email} \n'
        agenda.write(dados)
        agenda.close()
        print(f'Contato gravado com sucesso!!')
        msg=input('""\nDeseja mandar uma saudação nesse número?\nDigite 'S' para SIM.\nDigite 'N' para NÃO.\n:').upper()[0]
        cadastrado=True
    except:
        print('ERRO!! Contato NÃO gravado.')
    if cadastrado and msg=='S':
        f = datetime.datetime.now() + datetime.timedelta(seconds=61)
        kit.sendwhatmsg(f"+55{telefone.replace("-", "").replace(" ", "")}", f"Olá {nome}, seja bem-vindo na minha lista de conta
        menu()
```

# Listar Contato:

```
def ListarContato():  
    with open('agenda.txt', 'r') as f:  
        results = [[str(entry) for entry in line.split()] for line in f.readlines()]  
        results.sort()  
    for itens in results:  
        print(itens)
```

# Organizar Lista:

```
def organizarLista():  
    with open('agenda.txt', 'r') as agenda:  
        antes_de_organizar = agenda.readlines()  
        contato_organizados = sorted(antes_de_organizar, key=lambda x: x.split(";")[0])  
        print(contato_organizados)
```

```
if __name__ == '__main__':
```



# Deletar Contato:

```
def deletarContato():  
    nomeDeletado = input("Informe o nome para deletar o contato: ")  
    agenda = open ("agenda.txt", "r")  
    aux = []  
    aux2 = []  
    for i in agenda:  
        aux.append(i)  
    for i in range (0, len(aux)):  
        if nomeDeletado.upper() not in aux [i].upper():  
            aux2.append(aux[i])  
    agenda = open("agenda.txt", "w")  
    for i in aux2:  
        agenda.write(i)  
    print(f'Contato deletado com sucesso!!')  
    listarContato
```

# Buscar Contato:

```
def buscarContato():
    nome = input(f"Informe o nome a ser procurado: ").upper()
    agenda = open("agenda.txt", "r")
    encontrado = False
    for contato in agenda:
        # print(contato.split(";")[0].upper())
        if nome in contato.split(";")[0].upper():
            encontrado=True
            print(contato)
    agenda.close()
    if encontrado == False:
        print("\n\33[1;31;40mContato não encontrado. Tente novamente!\33[m\n")
        buscarContato()
```


# Checar E-mail:

```
def checar_email(email):  
    email_valido = False  
    provedores = ["gmail.com", "yahoo.com", "ymail.com", "hotmail.com", "outlook.com"]  
    provedor = email.split("@")[-1]  
  
    if provedor in provedores:  
        email_valido = True  
    return email_valido
```



# *Sair:*


```
def sair():  
    print(f'Até a próxima!')  
    exit()
```





# *Dificuldades:*

Programação do horário  
para o envio da mensagem para o  
whatsapp e e-mail.



# *Feito por:*

Graziella Alves

Isabelly Cristini

Thomas Laureano

Diana Santana

The image features a white background with decorative pink arcs in the top-left and bottom-right corners. The text "Thank You!!" is centered in a pink, cursive font.

*Thank  
You!!*