



INSTITUTO SUPERIOR POLITÉCNICO INTERCONTINENTAL DE LUANDA

DEPARTAMENTO DE ENGENHARIA

CURSO DE INFORMÁTICA

2º Ano/Turno: Manhã/Turma: EIM2

AGENDA TELEFÓNICA

Por:

Grupo D

DOCENTE

ANTUNES VITI

Ano académico/2023-2024



INSTITUTO SUPERIOR POLITÉCNICO INTERCONTINENTAL DE LUANDA

DEPARTAMENTO DE ENGENHARIA

CURSO DE INFORMÁTICA

2º Ano/Turno: Manhã/Turma: EIM2

AGENDA TELEFÓNICA

Trabalho de carácter avaliativo a ser entregue ao
docente da cadeira de Linguagem de Programação.

Docente:

Lic. **Antunes Viti**

Ano académico/2023-2024

INTEGRANTES

INTEGRANTES				
Nº	NOME COMPLETO	E-mail	NOTA DO TRABALHO	NOTA DA DEFESA
01	Aguinaldo Lencastre	clevandrolencastre@gmail.com		
02	António Alberto	antonioalberto14@mail.com		
03	João Augusto	johnnguifa6@gmail.com		
04	Sulamita Fitiboa	sulamitafitiboa@gmail.com		
05	Simão Guimarães	simondahollywood@gmail.com		
06	William Kahumba	willianngonga@gmail.com		
07	Edgar Luta	edgarluta63@gmail.com		

INTRODUÇÃO

O presente trabalho tem como finalidade debruçar sobre Agenda telefônica. Sendo que lista telefônica ou Agenda telefônica é uma publicação destinada à divulgação de informações sobre assinantes do serviço de telefonia. Basicamente uma lista telefônica é composta de três seções principais: lista de assinantes (por ordem de nomes), lista classificada (por produtos e atividades) e lista de endereços (por logradouros).

PARTE PRÁTICA DO PROJECTO

Inventado o telefone em 1876 e disseminado o seu uso, logo se fez sentir a necessidade por parte de quem o possuía da relação de quem também o possuísse.

Passando rapidamente na explicação do projeto. Vale expressar que o nosso projeto, é constituído por três (3) classes associadas entre si, que poderão compor a Agenda telefônica como já foi referido anteriormente.

A primeira classe é denominada Endereco, tendo como atributos: rua, casa, município, província e país.

```
class Endereco:
    def __init__(self, rua, casa, municipio, provincia, pais):
        self.rua = rua
        self.casa = casa
        self.municipio = municipio
        self.provincia = provincia
        self.pais = pais
```

A segunda classe é denominada Contato, tendo como atributos: nome, telefone, e-mail, endereço, e bi.

```
class Contato:
    def __init__(self, nome, telefone, email, bi, endereco):
        self.nome = nome
        self.telefone = telefone
        self.email = email
        self.endereco = endereco
        self.bi = bi
```

E a terceira e última é a classe Agenda, nessa classe, esta classe que pegará os atributos das classes anteriores, sendo parte dela, pois um contacto ou um endereço não tem a sua agenda, porém, a agenda contém contactos e endereços. Daí a relação das classes anteriores.

Não obstante destacarmos as classes que compõem a agenda, a mesma dispõe-se de um menu inicial, que por, e através dele são iniciados a cada uma das opções da agenda. Tais como:

- 1- Adicionar contato;
- 2- Editar contato;
- 3- Remover contato;

- 4- Buscar contato por nome;
- 5- Buscar contato por telefone;
- 6- Sair

Mais do que falar, precisamos demonstrar à prática. Assim sendo, deixamos abaixo o código como anexo, composto por 119 linhas de códigos. Passando para a apresentação do seu funcionamento.

```
print("=====")
print("                      AGENDA TELEFÔNICA                      ")
print("=====\\n")
class Endereco:
    def __init__(self, rua, casa, municipio, provincia, pais):
        self.rua = rua
        self.casa = casa
        self.municipio = municipio
        self.provincia = provincia
        self.pais = pais

class Contato:
    def __init__(self, nome, telefone, email, bi, endereco):
        self.nome = nome
        self.telefone = telefone
        self.email = email
        self.bi = bi
        self.endereco = endereco

class Agenda:
    def __init__(self):
        self.contatos = []

    def adicionar_contato(self, contato):
        self.contatos.append(contato)

    def editar_contato(self, nome, novo_contato):
        for i in range(len(self.contatos)):
            if self.contatos[i].nome == nome:
                self.contatos[i] = novo_contato
                break

    def remover_contato(self, nome):
        for contato in self.contatos:
            if contato.nome == nome:
                self.contatos.remove(contato)
                break

    def buscar_contato_por_nome(self, nome):
        for contato in self.contatos:
            if contato.nome == nome:
                return contato
        return None

    def buscar_contato_por_telefone(self, telefone):
        for contato in self.contatos:
            if contato.telefone == telefone:
                return contato
        return None

def menu_inicial():
    print("Apresentamos-lhe a agenda telefonica | Escolha as opções abaixo para começar a usar\\n")
    print("1. Adicionar contato")
    print("2. Editar contato")
```

```

print("3. Remover contato")
print("4. Buscar contato por nome")
print("5. Buscar contato por telefone")
print("6. Sair")
opcao = int(input("Escolha uma opção: "))
return opcao

# Exemplo de uso do menu
agenda = Agenda()
while True:
    opcao = menu_inicial()
    if opcao == 1:
        # Adicionar contato
        nome = input("Nome do contato: ")
        telefone = input("Telefone do contato: ")
        email = input("Email do contato: ")
        bi=input("BI ")
        rua = input("Rua: ")
        casa = input("Casa: ")
        municipio = input("Município: ")
        provincia = input("Província: ")
        pais = input("País: ")
        endereco = Endereco(rua, casa, municipio, provincia, pais)
        novo_contato = Contato(nome, telefone, email, bi, endereco)
        agenda.adicionar_contato(novo_contato)
    elif opcao == 2:
        # Editar contato
        nome = input("Nome do contato a ser editado: ")
        novo_nome = input("Novo nome do contato: ")
        novo_telefone = input("Novo telefone do contato: ")
        novo_email = input("Novo email do contato: ")
        novo_rua = input("Nova rua: ")
        novo_casa = input("Nova casa: ")
        novo_bi=input("Novo BI ")
        novo_municipio = input("Novo município: ")
        novo_provincia = input("Nova província: ")
        novo_pais = input("Novo país: ")
        novo_endereco = Endereco(novo_rua, novo_casa, novo_municipio, novo_provincia, novo_pais)
        novo_contato = Contato(novo_nome, novo_telefone, novo_email, novo_endereco)
        agenda.editar_contato(nome, novo_contato)
    elif opcao == 3:
        # Remover contato
        nome = input("Nome do contato a ser removido: ")
        agenda.remover_contato(nome)
    elif opcao == 4:
        # Buscar contato com o nome
        nome = input("Nome do contato a ser buscado: ")
        contato_encontrado = agenda.buscar_contato_por_nome(nome)
        if contato_encontrado:
            print(f"Contato encontrado: {contato_encontrado.nome} - {contato_encontrado.telefone}")
        else:
            print("Contato não encontrado")
    elif opcao == 5:
        # Buscar contato com nª de telefone
        telefone = input("Telefone do contato a ser buscado: ")
        contato_encontrado = agenda.buscar_contato_por_telefone(telefone)
        if contato_encontrado:
            print(f"Contato encontrado: {contato_encontrado.nome} - {contato_encontrado.telefone}")
        else:
            print("Contato não encontrado")
    elif opcao == 6:
        # para finalizar o programa
        break
    else:
        print

```

CONCLUSÃO

Durante a elaboração do trabalho, pelas pesquisas desenvolvidas, convenientemente concluímos que uma agenda telefônica bem preenchida com todos os elementos necessários, permite a fácil detenção e localização de indivíduos listado à nossa Agenda de contactos.