

"""

O objetivo deste programa é o mesmo da tarefa 1.
Use como base o código desenvolvido na tarefa 1.
Altere o código gerado na tarefa01, para utilizar o dicionário d_covid = {}.
Armazene os dados neste dicionário, onde a chave será o objeto Estado e o valor será uma lista com os objetos Cidades instanciados.

Menu

0- Finalizar o Programa

1- Cadastrar Estados

2- Cadastrar Cidades

3- Relatório de Estados

4- Relatório de Cidades

5- Atualizar números de casos Escolha:

Para armazenar os dados, você irá utilizar o dicionário d_codiv, para armazenar os objetos instanciados pela classe Estado e da classe Cidade

"""

```
class Estado:
    def __init__(self, nome_estado, sigla_estado):
        self.__nome_estado = nome_estado
        self.__sigla_estado = sigla_estado
        self.__cidade_estado = []

    def __str__(self):
        return f" {self.__nome_estado} "

    def adic_cidade_estado(self, nova_cidade):
        self.__cidade_estado.append(nova_cidade)

    @property
    def nome_estado(self):
        return self.__nome_estado

    @nome_estado.setter
    def nome_estado(self, nome):
        self.__nome_estado = nome

    @property
    def sigla(self):
        return self.__sigla_estado

    @sigla.setter
    def sigla(self, sigla):
        self.__sigla_estado = sigla

    @property
    def cidade_estado(self):
        return self.__cidade_estado

class Cidade:
    def __init__(self, nome_cidade):
        self.__nome_cidade = nome_cidade
        self.__qt_casos_cidade = 0

    def __str__(self):
        return f" {self.__nome_cidade} "

    def __int__(self):
        return {self.__qt_casos_cidade}

    @property
    def nome_cidade(self):
        return self.__nome_cidade

    def incluir_casos_cidade(self, casos):
        self.__qt_casos_cidade += casos

    @property
    def casos(self):
        return self.__qt_casos_cidade

d_covid = {}

def cadastrar_estados():
    nome_estado_ok = validar_nome_estado_ok().upper()
    sigla_estado_ok = validar_sigla_estado_ok().upper()
    d_covid.setdefault(Estado(nome_estado_ok, sigla_estado_ok), [])

def validar_nome_estado_ok():
```

```

while True:
    nome = input(f"Qual o Estado deseja cadastrar: ").upper()
    if not existe_nome_na_lista_estado(nome) and len(nome) > 0:
        if all(caract.isalpha() or caract.isspace() for caract in nome):
            break
        else:
            print("Por favor digite um nome válido (somente letras e espaços)")
            input(f"Esse Estado já existe ou valor digitado é nulo... Digite outro nome, Enter")
    return nome

```

```

def existe_nome_na_lista_estado(nome):
    for estado in d_covid.keys():
        if nome == estado.nome_estado:
            return True
    return False

```

```

def validar_sigla_estado_ok():
    while True:
        sigla = input(f"Qual a sigla deste Estado(OBRIGATÓRIO): ").upper()
        if not existe_sigla_na_lista_estado(sigla) and len(sigla) > 1 <= 2:
            if all(caract.isalpha() or caract.isspace() for caract in sigla):
                break
            else:
                print("Por favor digite um nome válido (somente letras e espaços)")
                input(f"Essa sigla já existe, ou possui Valor Nulo... Digite outra sigla, Enter")
    return sigla

```

```

def existe_sigla_na_lista_estado(sigla):
    for s in d_covid.keys():
        if sigla == s.sigla:
            return True
    return False

```

```

def identificar_estado_cidade():
    while True:
        estado_escolhido = input("Digite o estado para cadastrada essa cidade: ").upper()
        if not estado_escolhido:
            break
        for estado_na_lista in d_covid.keys():
            if estado_escolhido == estado_na_lista.nome_estado:
                print(f"Ok... Estado localizado...")
                return estado_na_lista

```

```

def cadastrar_cidade():
    cont = 0
    cidade_ok = identificar_cidade()
    cont += 1
    try:
        d_covid[identificar_estado_cidade()].append(Cidade(cidade_ok))
        if cont == 0:
            pass
    except:
        input("Cidade não cadastrada... Sem Estado Associado... ENTER")

```

```

def identificar_cidade():
    while True:
        cidade = input("Digite o nome da cidade para ser incluída a um Estado:").upper()
        if not existe_nome_na_lista_cidade(cidade) and len(cidade) > 0:
            if all(caract.isalpha() or caract.isspace() for caract in cidade):
                break
            else:
                print("Por favor digite um nome válido (somente letras e espaços)")
                input(f"Esta cidade já existe ou valor digitado é nulo...ENTER")
    return cidade

```

```

def existe_nome_na_lista_cidade(nome):
    for cidade in d_covid.values():
        for c in cidade:
            if nome == c.nome_cidade:
                return True
    return False

```

```

def cidade_quant_casos_ok():
    while True:
        nome_cidade = input(f"Digite o nome da cidade para incluir casos:").upper()
        if not nome_cidade:
            break
        for city in d_covid.values():
            for cidade in city:

```

```

        if nome_cidade == cidade.nome_cidade:
            print(f"Ok, Cidade localizada...{cidade}")
            caso = (input(f"Qual a quantidade de casos nesta Cidade: "))
            if caso.isdigit():
                caso = int(caso)
                if caso >= 0:
                    cidade.incluir_casos_cidade(caso)
                    print(f"{cidade} - {caso} casos")
                else:
                    input(f"Só são aceitos valores positivos...")
            break

```

```

def imprimir_lst_estados():
    print(f" Lista de Estados Cadastrados...")
    for est in d_covid:
        print(f"{est}")

```

```

def relatorio_estado():
    print(f" ===== Relatório dos Estados:")
    for estado, l_city in d_covid.items():
        total_casos = 0
        for cidade in l_city:
            total_casos += cidade.casos
        print(f"--->{estado} - Total de casos: {total_casos}")
    input(f"[Enter] Retorna ao Menu.")

```

```

def relatorio_cidade():
    print(f" ===== Relatório de Cidades:")
    for cidade in d_covid.values():
        for city in cidade:
            print(f'--->{city} - Casos Registrados: {city.casos}')
    input(f"[Enter] Retorna ao Menu.")

```

```

def imprimir_d_covid():
    for est, city in d_covid.items():
        print(f"{est} - {city}\n")

```

```

def interface():
    teclado = input(''
        MENU
        0 - Finalizar Programa
        1 - Cadastrar Estados
        2 - Cadastrar Cidades
        3 - Relatório de Estados
        4 - Relatório de Cidades
        5 - Atualizar números de casos
    Escolha:

```

```

    >>>: '')
    if teclado == "0":
        return
    if teclado == "1":
        cadastrar_estados()
    if teclado == "2":
        cadastrar_cidade()
    if teclado == "3":
        relatorio_estado()
    if teclado == "4":
        relatorio_cidade()
    if teclado == "5":
        cidade_quant_casos_ok()
    interface()

```

```

interface()

```