



MAPA DE NAVEGACION

INICIO

CREAR sistemaComedor

MIENTRAS True HACER

MOSTRAR "MENÚ PRINCIPAL"

MOSTRAR "1. Registrar beneficiario"

MOSTRAR "2. Registrar comedor"

MOSTRAR "3. Asignar cupo"

MOSTRAR "4. Registrar asistencia"

MOSTRAR "5. Generar reporte diario"

MOSTRAR "0. Salir"

LEER opcion

SI opcion == "1" ENTONCES

PEDIR nombre

PEDIR documento

SI beneficiario EXISTE documento EN sistemaComedor ENTONCES

MOSTRAR "Beneficiario ya registrado."

SINO

PEDIR edad

PEDIR poblacion_vulnerable (sí/no)

CONVERTIR poblacion_vulnerable A booleano
CREAR beneficiario CON datos
AGREGAR beneficiario A sistemaComedor
MOSTRAR "Beneficiario registrado."
FIN SI

SINO SI opcion == "2" ENTONCES
PEDIR nombre_comedor
SI comedor EXISTE nombre_comedor EN sistemaComedor ENTONCES
MOSTRAR "Comedor ya registrado."
SINO
PEDIR cupos_diarios
CREAR comedor CON datos
AGREGAR comedor A sistemaComedor
MOSTRAR "Comedor registrado."
FIN SI

SINO SI opcion == "3" ENTONCES
PEDIR documento_beneficiario
SI beneficiario NO EXISTE documento_beneficiario EN sistemaComedor ENTONCES
MOSTRAR "Beneficiario no encontrado."
SINO
PEDIR nombre_comedor
SI comedor NO EXISTE nombre_comedor EN sistemaComedor ENTONCES
MOSTRAR "Comedor no encontrado."
SINO
SI comedor TIENE cupos DISPONIBLES ENTONCES
ASIGNAR beneficiario AL comedor
MOSTRAR "Cupo asignado."
SINO
MOSTRAR "No hay cupos disponibles."
FIN SI
FIN SI
FIN SI

SINO SI opcion == "4" ENTONCES
PEDIR documento_beneficiario
SI beneficiario EXISTE documento_beneficiario EN sistemaComedor ENTONCES
REGISTRAR asistencia CON fecha_actual
MOSTRAR "Asistencia registrada."
SINO
MOSTRAR "Beneficiario no encontrado."
FIN SI

SINO SI opcion == "5" ENTONCES
PEDIR nombre_comedor
SI comedor EXISTE nombre_comedor EN sistemaComedor ENTONCES
MOSTRAR "Listado diario de", nombre_comedor
PARA CADA beneficiario EN comedor HACER
MOSTRAR beneficiario
FIN PARA

SINO

MOSTRAR "Comedor no encontrado."

FIN SI

SINO SI opcion == "0" ENTONCES

TERMINAR PROGRAMA

SINO

MOSTRAR "Opción inválida."

FIN SI

FIN MIENTRAS

FIN

JUSTIFICACION

El código implementa un sistema básico para gestionar beneficiarios y comedores comunitarios, permitiendo registrar personas, administrar cupos diarios, registrar asistencias y generar reportes.

Se utiliza un enfoque orientado a objetos con clases bien definidas (Beneficiario, Comedor, SistemaComedor) para mantener una estructura clara, modular y fácil de mantener. Se incluyen validaciones para evitar duplicados y asegurar integridad de datos.

La interfaz de menú por consola es simple, accesible y adecuada para usuarios sin experiencia técnica. El sistema es escalable, lo que permite futuras mejoras como reportes avanzados o almacenamiento en base de datos.

En resumen, el código es claro, funcional y está bien organizado para cumplir los objetivos de gestión en un comedor social.

CAPTURAS

comedor.py > ...

```
1  from datetime import date
2
3  class Beneficiario:
4      def __init__(self, nombre, documento, edad, poblacion_vulnerable):
5          self.nombre = nombre
6          self.documento = documento
7          self.edad = edad
8          self.poblacion_vulnerable = poblacion_vulnerable
9          self.asistencias = []
10
11     def registrar_asistencia(self, fecha):
12         if fecha not in self.asistencias:
13             self.asistencias.append(fecha)
14
15     def __str__(self):
16         return f"{self.nombre} ({self.documento})"
17
18
19 class Comedor:
20     def __init__(self, nombre, cupos_diarios):
21         self.nombre = nombre
22         self.cupos_diarios = cupos_diarios
23         self.beneficiarios_asignados = []
24
25     def asignar_beneficiario(self, beneficiario):
26         if len(self.beneficiarios_asignados) < self.cupos_diarios:
27             self.beneficiarios_asignados.append(beneficiario)
28             return True
29         return False
30
31     def listado_diario(self):
32         return [str(b) for b in self.beneficiarios_asignados]
33
34
35 class SistemaComedor:
36     def __init__(self):
37         self.beneficiarios = {}
38         self.comedores = {}
39
40     def registrar_beneficiario(self):
41         nombre = input("Nombre: ")
42         documento = input("Documento: ")
43         if documento in self.beneficiarios:
44             print("Beneficiario ya registrado.")
45             return
46         edad = int(input("Edad: "))
47         poblacion_vulnerable = input("Población vulnerable (sí/no): ").lower() == "sí"
48         self.beneficiarios[documento] = Beneficiario(nombre, documento, edad, poblacion_vulnerable)
49         print("Beneficiario registrado.")
50
51     def registrar_comedor(self):
52         nombre = input("Nombre del comedor: ")
53         if nombre in self.comedores:
54             print("Comedor ya registrado.")
55             return
56         cupos = int(input("Cupos diarios: "))
57         self.comedores[nombre] = Comedor(nombre, cupos)
58         print("Comedor registrado.")
```

```

60     def asignar_cupo(self):
61         doc = input("Documento del beneficiario: ")
62         if doc not in self.beneficiarios:
63             print("Beneficiario no encontrado.")
64             return
65         comedor_nombre = input("Nombre del comedor: ")
66         if comedor_nombre not in self.comedores:
67             print("Comedor no encontrado.")
68             return
69         comedor = self.comedores[comedor_nombre]
70         beneficiario = self.beneficiarios[doc]
71         if comedor.asignar_beneficiario(beneficiario):
72             print("Cupo asignado.")
73         else:
74             print("No hay cupos disponibles.")
75
76     def registrar_asistencia(self):
77         doc = input("Documento del beneficiario: ")
78         if doc in self.beneficiarios:
79             self.beneficiarios[doc].registrar_asistencia(str(date.today()))
80             print("Asistencia registrada.")
81         else:
82             print("Beneficiario no encontrado.")
83
84     def generar_reporte_diario(self):
85         comedor_nombre = input("Nombre del comedor: ")
86         if comedor_nombre not in self.comedores:
87             print("Comedor no encontrado.")

```

```

88         return
89         print(f"\nListado diario de {comedor_nombre}:")
90         for b in self.comedores[comedor_nombre].listado_diario():
91             print(f"-- {b}")
92
93     def menu(self):
94         opciones = {
95             "1": self.registrar_beneficiario,
96             "2": self.registrar_comedor,
97             "3": self.asignar_cupo,
98             "4": self.registrar_asistencia,
99             "5": self.generar_reporte_diario,
100            "0": exit
101        }
102        while True:
103            print("\n--- MENÚ ---")
104            print("1. Registrar beneficiario")
105            print("2. Registrar comedor")
106            print("3. Asignar cupo")
107            print("4. Registrar asistencia")
108            print("5. Generar reporte diario")
109            print("0. Salir")
110            opcion = input("Seleccione una opción: ")
111            accion = opciones.get(opcion)
112            if accion:
113                accion()
114            else:
115                print("Opción inválida.")

```

```

96         "2": self.registrar_comedor,
97         "3": self.asignar_cupo,
98         "4": self.registrar_asistencia,
99         "5": self.generar_reporte_diario,
100        "0": exit
101    }
102    while True:
103        print("\n--- MENÚ ---")
104        print("1. Registrar beneficiario")
105        print("2. Registrar comedor")
106        print("3. Asignar cupo")
107        print("4. Registrar asistencia")
108        print("5. Generar reporte diario")
109        print("0. Salir")
110        opcion = input("Seleccione una opción: ")
111        accion = opciones.get(opcion)
112        if accion:
113            accion()
114        else:
115            print("Opción inválida.")
116
117 if __name__ == "__main__":
118     sistema = SistemaComedor()
119     sistema.menu()
120

```