

Tarea 8

COLA DE PRIORIDAD

ACOTADA

Estructura de Datos

Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Aragón

REALIZADO POR:

**Enrique Emiliano Cano
García**

Clase ColaADT.

```
1 class ColaADT: 2 usages new *
2     def __init__(self): new *
3         self.cola = []
4
5     def esta_vacia(self): 6 usages new *
6         return len(self.cola) == 0
7
8     def longitud(self): 1 usage (1 dynamic) new *
9         return len(self.cola)
10
11    def frente(self): 2 usages new *
12        if self.esta_vacia():
13            return None # La cola está vacía
14        return self.cola[0]
15
16    def encolar(self, elem): 1 usage new *
17        self.cola.append(elem)
18
19    def des_encolar(self): 1 usage new *
20        if self.esta_vacia():
21            return None
22        return self.cola.pop(0)
23
24    def to_string(self): 1 usage new *
25        if self.esta_vacia():
26            return "Cola vacía"
27        return "Cola: " + " <- ".join(map(str, self.cola))
```

Clase Cliente

```
1 class Cliente: 9 usages 1 kno4
2     def __init__(self, nombre, perfil): 1 kno4
3         self.nombre = nombre
4         self.perfil = perfil
5
6     def __str__(self): 1 kno4
7         return f'Nombre: {self.nombre}, Perfil: {self.perfil}'
```

Clase Cola DePrioridad

```
1 from Tareas.Tarea7.ColaADT import ColaADT
2 from Tareas.Tarea8.Cliente_banco import Cliente
3 class ColaDePrioridad:
4     def __init__(self, max_prioridad):
5         self.max_prioridad = max_prioridad
6         self.colas = [ColaADT() for _ in range(max_prioridad + 1)]
7
8     def longitud(self):
9         total = 0
10        for cola in self.colas:
11            total += cola.longitud()
12        return total
13
14    def esta_vacia(self):
15        return self.longitud() == 0
16
17    def encolar(self, prioridad, elemento):
18        if 1 <= prioridad <= self.max_prioridad:
19            self.colas[prioridad].encolar(elemento)
20
21    def desencolar(self):
22        if self.esta_vacia():
23            print("No hay elementos")
24        else:
25            for cola in self.colas:
26                if not cola.esta_vacia():
27                    return cola.des_encolar()
28
29    def __str__(self):
30        result = []
31        for i, cola in enumerate(self.colas):
32            result.append(f"Prioridad {i}: {cola}")
33        return "\n".join(result)
34
35    # Mapeo de perfiles a prioridades
36    PERFILES_PRIORIDAD = {
37        "No es cliente": 5,
38        "Cliente nuevo": 4,
39        "Cliente frecuente": 3,
40        "Cliente premium": 2,
41        "Celebridad": 1
42    }
43
```

Main

```
44 def main(): 1 usage  ± kno4
45     cola = ColaDePrioridad(5)
46
47     cliente1 = Cliente("Cliente1", "Cliente nuevo")
48     cliente2 = Cliente("Cliente2", "Cliente nuevo")
49     cola.encolar(PERFILES_PRIORIDAD[cliente1.perfil], cliente1)
50     cola.encolar(PERFILES_PRIORIDAD[cliente2.perfil], cliente2)
51
52     cliente3 = Cliente("Cliente3", "No es cliente")
53     cliente4 = Cliente("Cliente4", "No es cliente")
54     cliente5 = Cliente("Cliente5", "No es cliente")
55     cola.encolar(PERFILES_PRIORIDAD[cliente3.perfil], cliente3)
56     cola.encolar(PERFILES_PRIORIDAD[cliente4.perfil], cliente4)
57     cola.encolar(PERFILES_PRIORIDAD[cliente5.perfil], cliente5)
58
59     celebridad = Cliente("Celebridad 1", "Celebridad")
60     cola.encolar(PERFILES_PRIORIDAD[celebridad.perfil], celebridad)
61
62     print("Estado de la cola actual: ")
63     print(cola)
64
65     siguiente_cliente = cola.desencolar()
66     if siguiente_cliente:
67         print(f"Atendiendo a {siguiente_cliente}, retirando $10,000")
```

```
68
69     cliente_frecuente = Cliente("Cliente 6", "Cliente frecuente")
70     cliente_premium = Cliente("Cliente 7", "Cliente premium")
71     cola.encolar(PERFILES_PRIORIDAD[cliente_frecuente.perfil], cliente_frecuente)
72     cola.encolar(PERFILES_PRIORIDAD[cliente_premium.perfil], cliente_premium)
73
74     siguiente_cliente = cola.desencolar()
75     if siguiente_cliente:
76         print(f"Atendiendo a {siguiente_cliente}")
77
78     print("Estado de la cola despues de atender algunos clientes: ")
79     print(cola)
80
81     while not cola.esta_vacia():
82         siguiente_cliente = cola.desencolar()
83         if siguiente_cliente:
84             print(f"Atendiendo a {siguiente_cliente}")
85
86     print("Estado final de la cola: ")
87     print(cola)
88
89 ► if __name__ == "__main__":
90     main()
91
```

Ejecución

```
Estado de la cola actual:
Prioridad 0: Cola vacía
Prioridad 1: Nombre: Celebridad 1, Perfil: Celebridad
Prioridad 2: Cola vacía
Prioridad 3: Cola vacía
Prioridad 4: Nombre: Cliente1, Perfil: Cliente nuevo, Nombre: Cliente2, Perfil: Cliente nuevo
Prioridad 5: Nombre: Cliente3, Perfil: No es cliente, Nombre: Cliente4, Perfil: No es cliente, Nombre: Cliente5, Perfil: No es cliente
Atendiendo a Nombre: Celebridad 1, Perfil: Celebridad, retirando $10,000
Atendiendo a Nombre: Cliente 7, Perfil: Cliente premium
Estado de la cola despues de atender algunos clientes:
Prioridad 0: Cola vacía
Prioridad 1: Cola vacía
Prioridad 2: Cola vacía
Prioridad 3: Nombre: Cliente 6, Perfil: Cliente frecuente
Prioridad 4: Nombre: Cliente1, Perfil: Cliente nuevo, Nombre: Cliente2, Perfil: Cliente nuevo
Prioridad 5: Nombre: Cliente3, Perfil: No es cliente, Nombre: Cliente4, Perfil: No es cliente, Nombre: Cliente5, Perfil: No es cliente
Atendiendo a Nombre: Cliente 6, Perfil: Cliente frecuente
Atendiendo a Nombre: Cliente1, Perfil: Cliente nuevo
Atendiendo a Nombre: Cliente2, Perfil: Cliente nuevo
Atendiendo a Nombre: Cliente3, Perfil: No es cliente
Atendiendo a Nombre: Cliente4, Perfil: No es cliente
Atendiendo a Nombre: Cliente5, Perfil: No es cliente
```

Estado final de la cola:

Prioridad 0: Cola vacía

Prioridad 1: Cola vacía

Prioridad 2: Cola vacía

Prioridad 3: Cola vacía

Prioridad 4: Cola vacía

Prioridad 5: Cola vacía

Process finished with exit code 0