

Universidad Americana

Facultad de Ingeniería y Arquitectura



Algoritmos y Estructuras de Datos, Grupo 4

Simulación de Páginas Web con Pilas

Realizado por:

Joaquín Alberto Pérez Zúñiga

Nicolás Rafael Laguna Vallejos

Docente:

MSc. César Marín López

1. `__init__.py`:

```
1  """
2  Ejercicio sobre colas.
3  """
4
5  from .cliente import Cliente
6  from .cola import Cola
7  from .sis_cine import SisCine
8
9
10 __all__ = [
11     "Cliente",
12     "Cola",
13     "SisCine"
14 ]
15
```

2. `sis_cine.py`:

```
1  """
2  Implementación de la clase SisCine, que maneja la simulación de la fila.
3  """
4
5  from . import Cliente, Cola
6
7
8  class SisCine:
9      """
10     Sistema para administrar una fila de clientes en un cine, utilizando una cola.
11     """
12
13     def __init__(self, fila: Cola[Cliente]) -> None:
14         self.fila_clientes = fila
15
16     def agregar_cliente(self, nuevo: Cliente) -> None:
17         """
18         Agrega un nuevo cliente al final de la fila/cola.
19         """
20
21         self.fila_clientes.enqueue(nuevo)
22
23     def atender_cliente(self) -> Cliente:
24         """
25         Atiende al próximo cliente y lo elimina de la cola.
26         """
27
28         return self.fila_clientes.dequeue()
29
```

3. cola.py:

```
1  """
2  Implementación de la clase Cola.
3  """
4
5  from typing import Generic, Iterable, Optional, TypeVar
6
7  T = TypeVar("T")
8
9
10 class Cola(Generic[T]):
11     """
12     Implementación personalizada de una cola.
13     """
14
15     def __init__(self, elementos: Optional[Iterable[T]] = None) -> None:
16         if elementos is None:
17             self.elementos = []
18         else:
19             self.elementos = list(elementos)
20
21     def __len__(self) -> int:
22         return len(self.elementos)
23
24     def __str__(self) -> str:
25         return str(self.elementos)
26
27     def is_empty(self) -> bool:
28         """
29         Comprueba si la cola esta vacía.
30         """
31         return len(self) == 0
32
33     def check_front(self) -> T:
34         """
35         Retorna el primer elemento de la cola.
36         """
37
38         return self.elementos[0]
39
40     def enqueue(self, elem: T) -> None:
41         """
42         Agrega un elemento a la cola.
43         """
44
45         self.elementos.append(elem)
46
47     def dequeue(self) -> T:
48         """
49         Elimina y retorna el primer elemento de la cola.
50         """
51
52         return self.elementos.pop(0)
53
```

4. cliente.py:

```
1  """
2  Implementación de la clase Cliente.
3  """
4
5  class Cliente:
6      """
7      Representa un cliente de un cine.
8      """
9
10     def __init__(self, pelicula: str, boleto: str):
11         self.pelicula = pelicula
12         self.boleto = boleto
13
14     def __str__(self) -> str:
15         return f"[ Película: {self.pelicula}; ID de Boleto: {self.boleto} ]"
16
17     def validar_boleto(self) -> bool:
18         """
19         Si el nombre de la película está en el boleto, es válido.
20         """
21
22         return self.pelicula in self.boleto
23
```

5. main.py:

```
1  """
2  La fila de clientes entrando a un cine se puede manejar utilizando una cola.
3  """
4
5  from os import system
6
7  from . import Cliente, Cola, SisCine
8
9
10 def menu_principal(sis: SisCine) -> None:
11     """
12     Loop principal del programa que le muestra un menú de opciones al usuario.
13     """
14
15     opcion = ""
16     while opcion != "4":
17         system("cls || clear")
18         print("\nSistema de Entrada para Cines")
19         print("-----\n")
20         print("1. Agregar cliente a fila")
21         print("2. Atender siguiente cliente")
22         print("3. Mostrar estado actual de la fila")
23         print("4. Salir")
24
25         opcion = input("\n-> ")
26
27         match opcion:
28             case "1":
29                 print("\nIngrese datos del cliente nuevo:")
30                 print("-----")
31                 sis.agregar_cliente(
32                     Cliente(
33                         input("Película que verá el cliente: "),
34                         input("ID del boleto del cliente: "),
35                     )
36                 )
37             case "2":
38                 print(f"Cliente atendido:\n{sis.atender_cliente()}")
39             case "3":
40                 input(f"Estado actual de la fila:\n{str(sis.fila_clientes)}")
41             case "4":
42                 input("\nSaliendo del programa...\n")
43             case _:
44                 input("\nOpción inválida, intente nuevamente!")
45
46
47 def main() -> None:
48     """
49     Ejecución del programa.
50     """
51
52     menu_principal(sis=SisCine(Cola()))
53
54
55 if __name__ == "__main__":
56     main()
57
```