



MATERIA: Algoritmos y Estructura de Datos

ALUMNO: Luciano Moliterno

TP 2024

## ENUNCIADO

Realizar un programa bajo el paradigma de TDA, que relacione un **mínimo de 3 estructuras** (sin contar las estructuras propias de listas, pilas y colas), en dicho programa tiene que existir un **menú** que permite ver en acción una **PILA** de algunas de las estructuras anidadas y además una **COLA** de otras estructuras. Es importante que sean creativos con el programa y que tenga sentido la implementación.

Ejemplo: “**un negocio tiene una cola de clientes y cada cliente tiene una pila de compras en el carro**”.

El programa tiene que tener una persistencia **básica**, por lo menos se debe guardar la información en un txt (si se puede recuperar aún mejor), si en vez de persistir en txt se animan a persistir en mysql.

## CONCEPTOS

- **Estructuras involucradas:**

- **Cliente:** Representa a un cliente del cine.
- **Snack:** Representa un producto que un cliente compra.
- **Sala:** Representa una sala del cine con información adicional.

- **Organización:**

- Una **cola** para gestionar a los clientes que ingresan al cine.
- Una **pila** para cada cliente, representando los snacks que compra.

- **Menú del programa:**

- Agregar un cliente a la cola.
- Añadir snacks a un cliente específico (pila).
- Mostrar la cola de clientes y sus snacks.
- Procesar al siguiente cliente (simula que ingresa al cine).
- Guardar o cargar los datos en un archivo.

## CARACTERÍSTICAS CLAVE Y PARADIGMA TDA

- **Cola:** Maneja a los clientes en el orden en que llegan.
- **Pila:** Cada cliente tiene una pila para almacenar los snacks que compra.
- **Persistencia:** Se guardan los datos de la cola y las pilas en un archivo de texto.
- **Archivo de encabezado (.h):**
  - Aquí defines la interfaz del TDA, es decir, las estructuras, clases, y las funciones (prototipos) que se usarán desde el exterior.
  - Es el contrato que otros archivos usarán para interactuar con el TDA.
- **Archivo de implementación (.cpp):**
  - Contiene las implementaciones de las funciones definidas en el archivo .h.
  - Este archivo hace "el trabajo sucio" del TDA, ocultando los detalles de implementación.
- **Archivo principal (main.cpp o main.c):**
  - Aquí se encuentra la función principal `main()` y se utiliza la interfaz del TDA (definida en el .h) para realizar las operaciones necesarias.
  - Este archivo no debe tener lógica compleja relacionada con el TDA, sino solo coordinar su uso.

## VENTAJAS

- **Modularidad:** Cada componente está claramente definido y separado.
- **Reutilización:** Puede usar los mismos TDAs en otros proyectos sin modificar el archivo principal.
- **Mantenimiento:** Los cambios son más fáciles de implementar, ya que están localizados.

## EVIDENCIA DE SALIDA

```
===== Menu Cine =====
1. Agregar cliente
2. Añadir snack a cliente actual
3. Mostrar cola de clientes
4. Procesar cliente
5. Guardar datos en archivo
6. Salir
Seleccione una opcion: _
```











```
Seleccione una opcion: 1
Ingrese el nombre del cliente: Luciano
```

```
Seleccione una opcion: 2
Ingrese el nombre del snack: 3D
Ingrese el precio del snack: 2500
```

```
Seleccione una opcion: 3
Cliente: Luciano
Snacks: 3D ($2500),
```

```
Seleccione una opcion: 4
Procesando cliente: Luciano
```

```
Seleccione una opcion: 5
Ingrese el nombre del archivo: Cliente_Luciano
Datos guardados en Cliente_Luciano
```

	bin	28/11/2024 11:41	Carpeta de archivos	
	obj	28/11/2024 11:41	Carpeta de archivos	
	Cine	28/11/2024 12:20	Archivo de origen ...	3 KB
	Cine	28/11/2024 12:23	Archivo de origen ...	1 KB
	Cliente_Luciano	28/11/2024 19:29	Archivo	0 KB
	main	28/11/2024 12:24	Archivo de origen ...	2 KB
	Snack	28/11/2024 12:19	Archivo de origen ...	1 KB
	Snack	28/11/2024 12:23	Archivo de origen ...	1 KB
	TP_2022_Cine.cbp	28/11/2024 11:39	Archivo CBP	2 KB
	TP_2022_Cine.depend	28/11/2024 19:27	Archivo DEPEND	1 KB

```
Seleccione una opcion: 6
Saliendo del programa...
```

```
Process returned 0 (0x0)   execution time : 282.46
4 s
Press any key to continue.
```