

Ejercicios estructuras enlazadas

Nuevo Spotify

1. **Crear una Playlist:** Define una estructura para representar una pila. Los elementos de la pila serán las canciones en tu playlist. Implementa funciones para añadir una canción (push), remover la última canción agregada (pop), y ver cuál es la última canción agregada (top).
 - El sistema debe poder mediante un menu:
 - Agregar canciones a la pila de reproducción (debe quedar primera)
 - Tomar la próxima canción para escucharla (removiendola de la pila)
 - Mostrar cual es la próxima canción (sin removerla)
2. **Revertir Playlist:** Agrega al sistema la posibilidad de revertir el orden de las canciones. Te recomendamos usar una pila auxiliar para implementarlo.
 - **Input:** ["Bohemian Rhapsody", "Hotel California", "Stairway to Heaven"]
 - **Output:** ["Stairway to Heaven", "Hotel California", "Bohemian Rhapsody"]
3. **Historial de Reproducción:** Ahora el cliente quiere tener un historial de reproducción. Para lo cual, cada vez que reproduces una canción, su titulo se debe guardar en una pila que representa el historial de reproducción y debe poder consumirse solo una vez.
4. **Historial de Reproducción v2:** El historial es un éxito entre los usuarios, por lo cual quieren agregarle nuevas funcionalidades.
 - a. Por cada registro del historial quiero saber:
 - i. Titulo de la canción
 - ii. El usuario debe calificar el tema luego de escucharlo, y esa calificación debe mostrarse en el historial (1 a 5 estrellas)
 - b. Quiero poder listar el historial ordenado por el orden en el que fue escuchado
 - c. Quiero decirle que me muestre filtrando por cantidad de estrellas.
 - i. **Input:** 1

- ii. **Output:** Todos los títulos de temas calificados con 1 estrella
- d. Quiero que me de la cantidad de estrellas promedio asignadas al historial de reproducción

Nuevo banco

1. **Cola del banco:** Implementa una estructura de datos para representar una cola en un banco. Añade funciones para ingresar un cliente a la cola (encolar), atender al próximo cliente (desencolar) y mostrar al próximo cliente en ser atendido (primero en la cola).
2. **Cola de prioridad:** Implementa una cola de prioridad en la cola del banco. Puedes, por ejemplo, dar prioridad a las personas mayores de 60 años.
3. **Registro de espera:** El banco quiere implementar un sistema para medir la espera de los clientes en el banco para saber cuando contratar más asistentes. Para lo cual, quieren tener un historial que diga la cantidad de personas que una persona tiene delante cada vez que se suma a una cola.

Torneo de futbol

1. **Cargar equipos:** Crea un sistema donde el usuario pueda cargar una lista de equipos. Cada nodo debe contener el nombre del equipo y los puntos que ha acumulado en el torneo hasta el momento. (Arrancan en 0)
2. **Asignar partidos a una cola:** El sistema debe permitir al usuario crear partidos con los equipos que están en la lista. Utiliza una cola para programar los partidos que se jugarán en el torneo. Cada elemento de la cola debe contener los nombres de los dos equipos que jugarán en ese partido.
3. **Jugar un partido pendientes:** El sistema debe poder jugar un partido eligiendo el ganador al azar. Debe tomar un partido de la cola, “jugar” el partido y asignar los puntos que corresponda al equipo en la lista.
4. **Encontrar al campeón:** Una vez que todos los partidos se han jugado y la cola está vacía, recorre la lista enlazada para encontrar al equipo con más puntos.

Compra de tickets

1. El sistema debe poder cargar una lista de espectáculos donde cada uno tiene:

- a. Nombre del artista
 - b. Cantidad de publico máximo
 - c. Cantidad de publico que compró el ticket
 - d. Cola de espera para comprar
2. El sistema debe poder sumar nuevas personas a la cola de espera correspondiente
 3. El sistema debe poder vender los tickets de un espectáculo hasta que la cola quede vacía o se quede sin tickets