PROYECTO PRÁCTICO NO. 01

Party Mix

Fecha de entrega: 18 de octubre de 2021

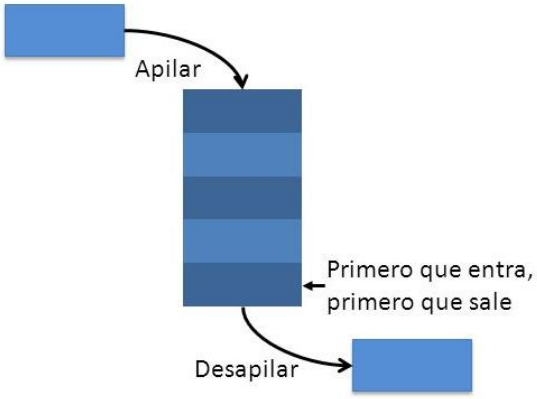
Descripción

La pandemia del COVID 19 realizó un cambio drástico en la metodología de cursos en línea como presenciales, se ha visto que muchos estudiantes han empezado a perder interés en las mismas. Sin embargo, se ha percatado que la música ha ayudado con el desinterés, permitiendo acompañar a los estudiantes en el transcurso de cada curso, reproducir música que ayuda a su estado de ánimo.

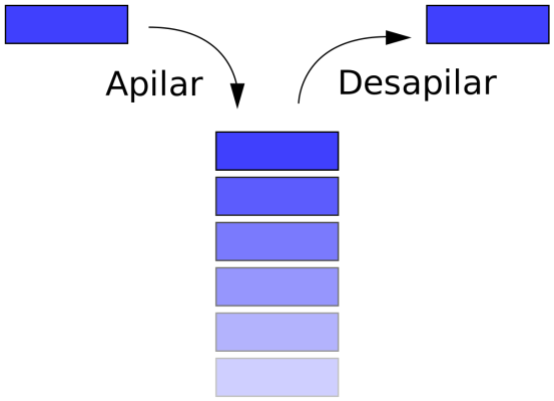
Por lo anterior, como futuros profesionales, se les ha proporcionado la tarea de crear un programa que permita a los estudiantes crear una *playlist* de música para reproducirla durante sus trabajos en grupo, individuales y principalmente en los proyectos de programación.

Enunciado

Las pilas y colas son herramientas que se utilizan día con día, se ven presentes en los supermercados, en el trafico y finalmente en las aplicaciones de uso diario.
El concepto de pila es organizar elementos encima de otro, para remover un elemento solo es permitido sacar el último elemento ingresado (en la cima).
El concepto de cola es colocar elementos en forma consecutiva, para remover un elemento es permitido sacar solamente el primer elemento ingresado (en el fondo).



COLA



PILA

En base a lo anterior, se solicita un simulador del proceso de *playlist*, empleando los conceptos de **estructura de datos, pilas, colas y ordenamientos.**
Para esto existe una *playlist* sin ningún orden en particular, en la cual se permitirá al usuario interactuar con ella, así mismo existirá una fila de reproducción que permita al usuario ingresar más canciones, para esto para esto existen las siguientes reglas:

Playlist - Pila

- Iniciar una **pila** - playlist como se muestra a continuación.

Canción - Artista
Playa – Nicky Jam
Te amo – Piso 21
Me olvide – Rels B
Ámate – Micro TDH
Limo – Jesse Baez
Túnel – Drefquila
Nubes – Nicole
Fuego – Nicole
Púrpura - Nanpa
Romance - Samantha

- Se podrá definir la *playlist* inicial por medio de un archivo csv, separado por comas, donde se encontrará cada canción con su artista del siguiente formato:
    - **Playa – Nicky Jam,** Te amo – Piso 21, **Me olvide – Rels B ,** Amate – Micro TDH
- La información anterior deberá almacenarse en tiempo real en un objeto con las características necesarias.
- Se debe de validar la entrada de la *playlist*, para que todas las canciones posean un artista, si en dado caso no existirá un artista definido debe de poder mostrar la palabra **desconocido.**
- El usuario debe de contar con la opción de cambiar el orden de la pila permitiéndola ordenar, ascendente o descendentemente, por **nombre de canción** o **artista**.
    - Al ordenar por artista, si el artista es **desconocido**, se debe colocar al final de la playlist.
- El usuario debe de contar con la opción de visualizar la canción en reproducción (la primera de la pila) y la *playlist* completa para saber cuales son las siguientes.
- El usuario puede tener la opción de reproducir, el cual sacará el valor disponible de la pila, cada vez que se reproduzca se deberá quitar la canción de la *playlist*.
- Si en dado caso ya no hay mas canciones para reproducir se debe de trasladar los datos de la fila de reproducción a la *playlist* **(mover la cola a la pila)**
- El usuario debe ser capaz de exportar en un archivo la *playlist* actual.
- Se debe mostrar en todo momento el estado de la pila.

Fila de Reproducción - Cola

- Iniciar una cola – fila de reproducción
    - Permitirá almacenar las canciones que el usuario desea reproducir en el futuro
- El usuario debe ser capaz de agregar una canción a la fila de reproducción.
    - Debe de solicitar nombre y artista para que tenga la misma estructura de la cola.
- El usuario debe ser capaz de eliminar una canción de la fila de reproducción.

- El usuario debe ser capaz de sincronizar la fila de reproducción a la *playlist* y mantener el orden seleccionado previamente (nombre o artista).
- Se debe mostrar en todo momento el estado de la cola.
- El usuario podrá reproducir la anterior o la siguiente canción.
    - La reproducción podrá ser secuencial o aleatoria, según elija el usuario

Entregables

- Archivo comprimido o repositorio en Github con:
    - Carpeta de Solución del Proyecto (código fuente con documentación interna)
    - Programa ejecutable (podrá omitirse si es repositorio de Github)
    - Documentación Externa

- La documentación externa contiene:
    1. Carátula
    2. Introducción
    3. Análisis
        a. Entradas
        b. Salidas
        c. Procesos
        d. Restricciones
    4. Diseño
        a. Diagrama de Flujo
    5. Conclusiones
    6. Recomendaciones
    7. Referencias
        a. Librerías utilizadas y su utilización
    8. Anexos
        a. Manual de Usuario

Aspectos para evaluar

- Validación de errores, uso de las técnicas de programación adecuadas para la resolución.
- Adecuada aplicación de los conocimientos vistos en clase
- Calidad de la documentación: ortografía, orden, limpieza y que esté completa.
- Calidad de la solución propuesta: que solucione el problema (que haga lo que requiere el enunciado) en forma eficaz.

- Funcionalidad del programa: debe cumplir a cabalidad con todos los requerimientos.
- Evidencia de la creación del programa y dominio de los conceptos utilizados.
- Creatividad.

Ponderación

Documentación	25
Codificación	75
Total	100

Consideraciones

- Se trabajará en modalidad individual.
- Se debe realizar una solución Windows Form, en lenguaje de programación C++.
- Deberá realizarse la **creación de estructuras propias, así como la utilización de punteros**.
- Toda solución presentada debe compilar correctamente para poder tener derecho a revisión.
- La utilización de código de terceros debe estar completamente documentado, referenciado y justificada su utilización, debe demostrarse el dominio completo de lo implementado.
- **Se podrá demandar que en la calificación presencial del proyecto se realicen cambios de funcionalidad.**