



UNIVERSITAT DE VALÈNCIA

Laboratorio de PROGRAMACIÓN

Grado en Ingeniería Multimedia (1º)

Curso 2021-22

Práctica Nº 8: Listas

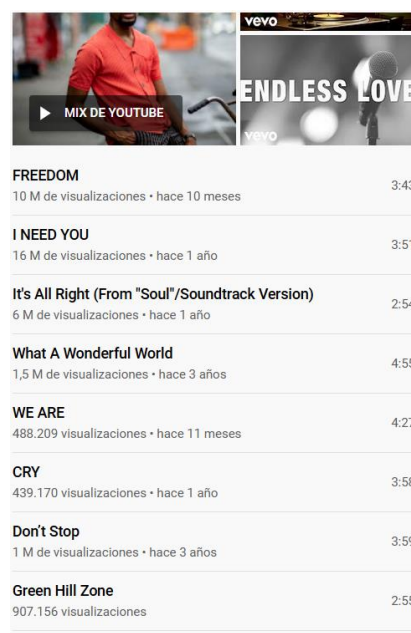
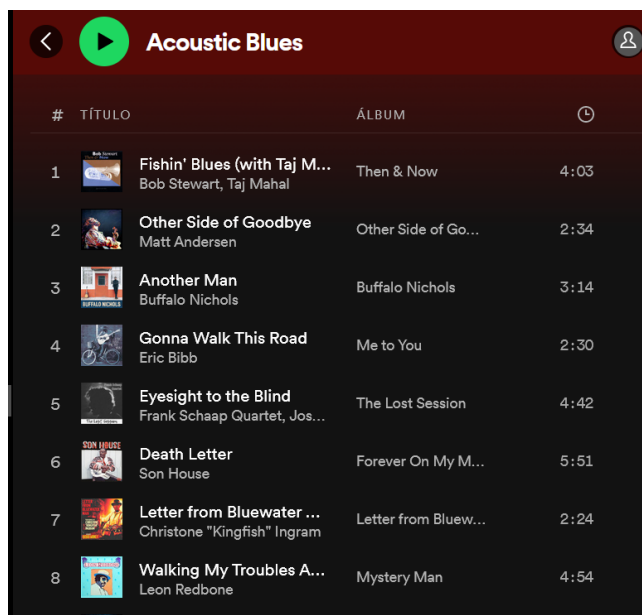
Periodo de realización: Semana del 16/05/2022 a 20/05/2022

Material a entregar

- (Repositorio) Proyecto compilable con el código fuente necesario para la práctica.
- (Tarea AV) Documentación del código generada por Doxygen.

Introducción

En esta práctica se pretende especificar y desarrollar una aplicación para gestionar una lista de reproducción de archivos multimedia de música (canciones o vídeos de música), semejante a la que se utiliza en aplicaciones comerciales (*Youtube, Spotify, Windows Media Player, etc.*).



La aplicación gestionará solo una lista de reproducción, en la que cada elemento (como se detalla más adelante) contendrá el título de la canción/vídeo, información complementaria (como intérprete, autor, etc.), la ubicación del archivo para su reproducción y la duración. Dentro de la lista de reproducción siempre habrá un archivo **'activo'**, entiendo por tal el archivo sobre el que se realizará la siguiente operación (reproducir, eliminar, etc.).

Ejercicios

Tarea 1

Para gestionar la lista de reproducción, haremos uso de un *TAD Tiempo* especializado en manejar la duración (horas, minutos y segundos) de cada archivo multimedia. La clase está en los ficheros ("Pr8_Tiempo.h", "Pr8_Tiempo.cpp"), que se suministran en el material de la práctica y que sigue la siguiente especificación:

TAD Tiempo

Dominio: representa la duración de los archivos especificando los atributos:

- *horas*: valor de la hora en el intervalo 0 a 23.
- *minutos*: valor de los minutos en el intervalo 0 a 59.
- *segundos*: valor de los segundos en el intervalo 0 a 59.

Operaciones:

- *CrearTiempo ()* -> *Tiempo*
- *CrearTiempo (Entero, Entero, Entero)* -> *Tiempo*
- *EstableceTiempo (Tiempo, entero, entero, real)* -> *Lógico*
- *DevuelveHora (Tiempo)* -> *Entero*
- *DevuelveMinuto (Tiempo)* -> *Entero*
- *DevuelveSegundo (Tiempo)* -> *Entero*
- *Mostrar (Tiempo)*
- *Leer(Tiempo)* -> *Tiempo*
- *Incrementar(Tiempo, Tiempo)* -> *Tiempo*
- *Decrementar(Tiempo, Tiempo)* -> *Tiempo*

Axiomas:

Sea $t \in \text{Tiempo}$, $h, m, s \in \text{Entero}$ se cumple que:

- *Tiempo ()* Construye el TAD *Tiempo*, e inicializa el valor de horas y minutos a 0 y segundos a 0.0.
- *Tiempo (h, m, s)* Construye el TAD *Tiempo*, e inicializa el valor de horas con h , minutos a m y segundos a s .
- *EstableceTiempo (t, h, m, s)* Si los valores son correctos, establece el valor h para las horas, el valor m para los minutos y el valor de s para los segundos y devuelve cierto en otro caso devuelve falso.
- *DevuelveHoras (t)* Devuelve el valor de las horas.
- *DevuelveMinutos (t)* Devuelve el valor de los minutos.
- *DevuelveSegundos (t)* Devuelve el valor de los segundos.
- *Mostrar (t)* Muestra el valor del tiempo.
- *Leer (t)* Lee por el valor del tiempo.
- *Incrementar(t, t1)* Incrementa el tiempo t con el tiempo $t1$
- *Decrementar(t, t1)* Decrementa el tiempo t con el tiempo $t1$

Las operaciones *Mostrar* y *Leer* se han implementado mediante la sobrecarga de los operadores "<<" y ">>" y las operaciones *Incrementar* y *Decrementar* mediante la sobrecarga de los operadores "+=" y "-=", respectivamente.

SCV: Debes trabajar en la carpeta “Pr8” del repositorio que ya tienes creado. Descarga en ella los archivos “Pr8_Tiempo.h” y “Pr8_Tiempo.cpp” que se proporcionan como material. Haz un primer *commit* (“Pr.8: Versión inicial”) y *push* con estos archivos.

Tarea 2

La lista de reproducción contendrá una serie de archivos multimedia (audio o vídeo), y para cada uno de ellos tendrá asociada información de su título, información complementaria sobre el Media, ruta del sistema donde esté guardado el archivo y duración (en horas, minutos y segundos).

Crea la clase Media en los ficheros “Pr8_Media.h” y “Pr8_Media.cpp” a partir de la siguiente especificación:

TAD Media

Dominio: conjunto de datos de un archivo multimedia:

- título
- información complementaria sobre el Media
- ruta
- duración (Tiempo)

Operaciones:

- *CrearMedia()* -> Media
- *EstablecerMedia (Media, Cadena, Cadena, Cadena, Tiempo)* -> Media
- *EstablecerTitulo(Media, Cadena)* -> Media
- *EstablecerInformacion(Media, Cadena)* -> Media
- *EstablecerRuta(Media, Cadena)* -> Media
- *EstablecerTiempo(Media, Tiempo)* -> Media
- *DevuelveTitulo(Media)* -> Cadena
- *DevuelveInformacion(Media)* -> Cadena
- *DevuelveRuta(Media)* -> Cadena
- *DevuelveTiempo(Media)* -> Tiempo
- *Leer(Media)* -> Media
- *Mostrar (Media)*

Axiomas:

Sea $m \in \text{Media}$, $e \in \text{Entero}$, $c1, c2, c3 \in \text{Cadena}$, $t \in \text{Tiempo}$, $l \in \text{Logico}$, se cumple que:

- | | |
|---|--|
| • <i>CrearMedia ()</i> | <i>Crea un media. Inicializa los atributos con valores por defecto.</i> |
| • <i>EstablecerMedia (m, c1, c2, c3, t)</i> | <i>Crea un media. Inicializa los atributos con los valores c1,c2,c3 y t.</i> |
| • <i>EstablecerTitulo(m, c1)</i> | <i>Guarda el título del archivo multimedia.</i> |
| • <i>EstablecerInformacion(m, c1)</i> | <i>Guarda información complementaria sobre el media.</i> |
| • <i>EstablecerRuta(m, c1)</i> | <i>Guarda la ruta del sistema donde esté guardado el archivo.</i> |
| • <i>EstablecerTiempo(m, t)</i> | <i>Guarda el tiempo de duración del archivo.</i> |
| • <i>DevuelveTitulo (m)</i> | <i>Devuelve el título del archivo multimedia.</i> |
| • <i>DevuelveInformacion(m)</i> | <i>Devuelve información del archivo multimedia.</i> |
| • <i>DevuelveRuta (m)</i> | <i>Devuelve la ruta al archivo multimedia.</i> |
| • <i>DevuelveTiempo (m)</i> | <i>Devuelve el tiempo de duración del archivo multimedia.</i> |

- *Leer (m)* *Obtiene la información de un archivo multimedia.*
- *Mostrar(m)* *Muestra la información de un archivo multimedia.*

Implementa la operación *Mostrar* mediante la sobrecarga del operador '>>' para asignar la información de un *Media* desde un stream (fichero, teclado, ...) y el operador '<<' para sacar la información de un *Media* por un stream (fichero, consola, ...).

SCV: Registra los cambios en de "Pr8_Media.h" y "Pr8_Media.cpp" y haz un (*commit*) con el mensaje "Pr8: Completada Tarea 2".

Tarea 3

En esta tercera tarea vamos a añadir a nuestro proyecto la clase lista. Para ello, y dado que no sabemos cuántos archivos multimedia tendremos, utilizaremos la implementación dinámica doblemente enlazada circular y con nodo cabeza que se encuentra implementada en los apuntes de la asignatura.

Escribe Pr8_ListaMedia.h y Pr8_ListaMedia.cpp y utilizando como tipo de datos Valor la clase *Media* y añádela al proyecto.

Además, vamos a **ampliar** el TAD Lista con la siguiente especificación:

TAD Lista

Dominio: a los atributos de la lista se añaden:

- Activo: número de orden del archivo activo,
- Cuantos: número total de archivos en la lista,
- Total: tiempo total de la lista de reproducción.

Operaciones:

- *Insertar(Lista, Valor, Entero)*
- *MoverAlFinal (Lista) -> Lógico*
- *Retroceder (Lista) -> Lógico*
- *MostrarPantalla (Lista)*
- *GuardarEnFichero(Lista, Cadena) -> Lógico*
- *RecuperarDeFichero(Lista, Cadena) -> Lógico*

Axiomas:

Sea $l \in \text{Lista}$, $v \in \text{Valor}$, $e \in \text{Entero}$, $c \in \text{Cadena}$ se cumple que:

- *Insertar(l, v, e)* $e = \text{Posición que sea, entre } 0 \text{ y } n$
Inserta un valor v en una posición e dada de la lista. posición respecto al activo en que se quiere insertar: 0 delante, 1 detrás.
- *MoverAlFinal (l)* *Mueve el elemento activo y lo pone al final de la lista.*
- *Retroceder (l)* *Retrocede el punto de atención al anterior elemento de la lista, de modo circular.*
- *MostrarPantalla (l)* *Muestra en pantalla el contenido de la lista (archivos multimedia), desde el principio al final.*
- *GuardarEnFichero(l, c)* *Guarda en el fichero de nombre c la lista la información almacenada en la lista multimedia.*

- *RecuperarDeFichero (l, c)* Guarda en la lista l la información almacenada en el fichero c.

SCV: Registra los cambios en de “Pr8_Lista.h” y “Pr8_Lista.cpp” y haz un (*commit*) con el mensaje “Pr8: Completada Tarea 3”.

Tarea 4

Una vez implementadas todas las clases necesarias y comprobado su funcionamiento, vamos a crear un programa para que gestionar los archivos multimedia.

Al ejecutar el programa, se mostrará un menú al usuario en el que cada vez que se seleccione una opción, se ejecutará y se volverá a mostrar el menú para seguir interactuando con los pacientes, hasta que el usuario pulse la opción 0. Las opciones que ofrecerá el menú son las siguientes:

1. **Añadir** un archivo multimedia **al final** de la lista, pidiéndole al usuario los datos del archivo que se va a añadir; si la lista estuviera previamente vacía el archivo añadido pasaría a ser el archivo activo.
2. **Añadir** un archivo multimedia **en una posición** determinada de la lista, pidiéndole al usuario los datos del archivo que se va a añadir e indicando (con un número) en qué posición de la lista debe añadirse; **si la posición indicada es mayor que la última de la lista, se añadirá al final**, y si no se añadirá en la posición indicada anteponiéndose al archivo que ocupará esa posición, pero sin suprimirlo.
3. **Eliminar un archivo** de la lista indicándolo por el número de orden que indica su posición; si el archivo indicado fuera el activo en ese momento el punto de interés pasaría al siguiente elemento de la lista.
4. **Mover el archivo activo al final de la lista**, dejando como archivo activo el que estaba situado a continuación del que era el archivo activo antes de este movimiento.
5. **Avanzar** el punto de interés (el archivo activo) al siguiente elemento de la lista; si el punto de interés era el último archivo de la lista, se pasará al primero.
6. **Retroceder** el punto de interés al anterior archivo de la lista; si el punto de interés era el primer archivo de la lista, se pasará al último.
7. **Guardar la lista** actual en un archivo, cuyo nombre se solicitará al usuario, para poder recuperarla en otro momento.
8. **Recuperar una lista** de reproducción a partir de un archivo cuyo nombre se solicitará al usuario¹.

Al iniciarse la aplicación, y también después de cada una de las acciones, la aplicación mostrará toda la información del archivo ‘activo’ en ese momento (el que se está reproduciendo), si es que hay algún archivo en la lista, y a continuación un listado con la información que se considere adecuada (a criterio de los estudiantes) de los archivos que componen la lista de reproducción. Para cada elemento de la

¹ En AV se proporciona un archivo de datos, “canciones.txt”.

lista se mostrará delante un número de orden secuencial que empezará por 1. Además, el archivo 'activo' aparecerá destacado (por ejemplo, con unos caracteres '>>' delante del número de orden).

La visualización podría ser como la que se muestra a continuación:

número de orden y número total

```
[2/3] Everybody Needs Somebody to Love    00:03:24
The Blues Brothers - Original Soundtrack
C:\Users\Usuario\Music\Everybody.mp3
```

*todos los atributos del
archivo 'activo'*

*destacando
el archivo
'activo'*

>>

```
1 She Caught the Katy                00:04:12
2 Everybody Needs Somebody to Love    00:03:24
3 Jailhouse Rock                     00:03:20
TIEMPO TOTAL DE REPRODUCCIÓN         00:10:56
```

*listado con parte
de la información
de cada archivo
de la lista*

```
Indica tu opcion (0-SALIR)
 1-Añadir al final,    2-Añadir antes/despues del activo,
 3-Eliminar el activo, 4-Mover al final,
 5-Avanzar,            6-Retroceder,
 7-Guardar,            8-Recuperar : 8
Nombre del archivo del que recuperar una lista: canciones.txt

=====
[1/40] Título: Harlem Nocturne
Duración: 0:05:06
Información: Jazz & New York Disc 1
Ruta en el sistema: (C:\Music\Jazz\Jazz & New York Disc 1\01 Harlem Nocturne.wma)
=====
>> 1. Harlem Nocturne                0:05:06
    2. I Remember Harlem              0:05:59
    3. Breakfast at Tiffany's        0:04:52
    4. Penthouse Serenade            0:05:55
    5. Bronx Blues                   0:02:42
    6. Central Park Swing            0:05:45
    7. New York 19                   0:04:40
    8. Park Avenue Petite            0:02:07
    9. Sax Fifth Avenue              0:02:58
   10. Lullaby Of Bridland            0:05:17
   11. Rose of Washington Square     0:05:12
   12. Rhapsody in Blue              0:02:15
   13. Harlem Joys                   0:04:09
   14. Park Avenue Escapade          0:02:41
   15. Chinatown,My Chinatown        0:05:42
   16. Broadway                      0:05:55
   17. Scrapple From the Apple       0:05:03
   18. Bohemia After Dark            0:04:17
   19. Central Park Blues            0:02:44
   20. Grand Central                 0:05:50
   21. Stand By Your Man             0:05:51
   22. The Shelter Of Your Arms      0:03:02
   23. Angel Of The Morning          0:04:47
   24. You're The Nearest Thing To Heaven 0:02:46
   25. Walkin' After Midnight        0:04:30
```

```

26. It's Four In The Morning          0:04:58
27. Ease The Want In Me              0:03:27
28. Please Help Me, I'm Falling      0:03:41
29. Crying                           0:05:23
30. Hey, Good Lookin'                0:04:11
31. Help Me Make It Through The Night 0:02:55
32. My Mountain Dew                  0:04:57
33. Crazy Baby                       0:05:47
34. Train Of Love                     0:04:34
35. For The Good Times                0:03:20
36. I Can't Stop Loving You           0:05:12
37. Crazy Arms                       0:04:04
38. Heaven's Just A Sin Away          0:02:01
39. Hungry For Love                  0:05:20
40. Jealous Heart                     0:02:01
                                     TIEMPO TOTAL 2:51:56
=====
Indica tu opcion (0-SALIR)
1-Añadir al final,    2-Añadir antes/despues del activo,
3-Eliminar el activo, 4-Mover al final,
5-Avanzar,            6-Retroceder,
7-Guardar,            8-Recuperar :

```

SCV: Registra los cambios en de "Pr8.cpp" y haz un (*commit*) con el mensaje "Pr8: Completada Tarea 4".

Tarea 5: Generación de documentación

El código realizado deberá estar correctamente documentado con Doxygen, de acuerdo con la guía de estilo. Una vez realizada la versión final del programa, se deberá ejecutar Doxygen para generar toda la documentación definitiva. La documentación deberá ser generada exclusivamente en formato en *html* (utilizar como referencia, convenientemente personalizado para esta práctica, el archivo de configuración de Doxygen disponible en el Aula Virtual). Solo la documentación deberá subirse a Aula Virtual (como archivo comprimido). **El programa NO se debe subir al AV, ya que será evaluado directamente desde el repositorio de control de versiones.**

La documentación generada con Doxygen NO debe subirse al repositorio de control de versiones Se debe mantener exclusivamente en el ordenador personal.