



SUBPROGRAMAS Y ESPECIFICACIÓN



TITLE: PROGRAMMING II LABS

SUBTITLE: Práctica 1

AUTHOR 1: Raúl Fernández del Blanco LOGIN 1: r.delblanco

AUTHOR 2: Armando Martínez Noya LOGIN 2: a.mnoya

Date: 09/04/21

En el programa principal del proyecto (*main.c*) encontraremos unas funciones las cuales son declaradas y definidas en él, estas funciones fueron diseñadas por los autores de este proyecto con la finalidad de conseguir el correcto funcionamiento de éste. Estas funciones son:

- **void New ();**
- **void Delete ();**
- **void Play ();**
- **void Stats ();**

Estos subprogramas ejecutan las ordenes indicadas en los ficheros de entrada de la siguiente manera:

void New (char *commandNumber, char command, char *param1, char *param2, tList *lista);

1. Objetivo:

Se encarga de recibir la información y llamar a las funciones pertinentes para crear el nuevo usuario con la información recibida.

2. Entradas:

- CommandNumber: indica el número de operación que se ejecuta.
- Command: indica la operación a ejecutar.
- Param1: indica el nickname del nuevo usuario.
- Param2: indica la categoria de dicho usuario.
- Lista: indica la lista donde se añadirá el nuevo usuario.

3. Salidas:

La lista con un nuevo usuario implementado en la última posición.

4. PreCondición:

- La lista tiene que estar previamente inicializada.
- El param2 debe ser válido.

5. PostCondición:

La última posición de la lista pasará a ser la del último usuario añadido.

void Delete (char *commandNumber, char command, char *param1, tList *lista);

1. Objetivo:

Se encarga de eliminar de la lista el usuario que tenga el nickname indicado.

2. Entradas:

- CommandNumber: indica el número de operación que se ejecuta.
- Command: indica la operación a ejecutar.
- Param1: indica el nickname del usuario a eliminar.
- Lista: indica la lista donde se eliminará el usuario.

3. Salidas:

La lista modificada sin el elemento borrado anteriormente.

4. PreCondición:

La lista tiene que estar previamente inicializada.

5. PostCondición:

Las posiciones de los elementos de la lista posteriores a la del elemento insertado han podido variar.

void Play (char *commandNumber, char command, char *param1, char* param2, tList *lista);

1. Objetivo:

Es el programa que se encarga de aumentar la cantidad de reproducciones que tiene un usuario y mostrar lo que está reproduciendo.

2. Entradas:

- CommandNumber: indica el número de operación que se ejecuta.
- Command: indica la operación a ejecutar.
- Param1: indica el nickname del usuario.
- Param2: indica el nombre del video a reproducir.
- Lista: indica la lista en la que se modificarán las reproducciones.

3. Salidas:

La lista actualizada con las reproducciones del usuario introducido aumentadas en una unidad.

4. PreCondición:

La lista tiene que estar previamente inicializada.

void Stats (char *commandNumber, char command, tList lista);

1. Objetivo:

Tiene como objetivo calcular y mostrar por pantalla de manera ordenada las estadísticas de la lista.

2. Entradas:

- CommandNumber: indica el número de operación que se ejecuta.
- Command: indica la operación a ejecutar.
- Lista: indica la lista a la que pertenecen los datos.

3. Salidas:

Tiene como salida por pantalla la lista completa de usuarios actuales, así como el número de usuarios, el número de vídeos reproducidos y la media de reproducciones para cada categoría.

4. PreCondición:

La lista tiene que estar previamente inicializada.

Además, en nuestro programa principal podremos encontrarnos con tres funciones las cuales se encargan de leer los ficheros de entrada y llamar a nuestros subprogramas definidos anteriormente. Estas son:

int main (int nargs, char **args);

La función main de nuestro programa se encarga de seleccionar y enviar el documento que leerá posteriormente.

void readTasks (char *filename);

La función recibe como entrada el fichero que queremos que utilice el programa. Esta, además de inicializar la lista, se encarga de leer línea a línea dicho fichero y extraer las variables necesarias para el correcto funcionamiento del programa. Las variables mencionadas anteriormente, son las introducidas en la siguiente función:

void processCommand (char *commandNumber, char command, char *param1, char *param2, tList *list);

Esta función es muy sencilla, ya que simplemente se encarga de elegir cuál de los subprogramas previamente analizados se va a ejecutar y de enviarles la información pertinente para su correcto funcionamiento.