

Práctica 4:

Aplicación *RedPelis*

Fecha de entrega: 10 de mayo de 2012

1. Descripción de la práctica

Hemos recibido el encargo de preparar un programa para la gestión de los datos de una empresa de alquiler de películas en *streaming*, llamada RedPelis. La empresa dispone de un catálogo de películas y una base de datos de clientes. Los clientes pueden conectarse a la aplicación y ver una película en *streaming* directamente en su ordenador, pagando un precio por el visionado. Al final de la película, el cliente puede indicar si le ha gustado o no, para que el sistema pueda mantener estadísticas sobre los gustos de sus clientes.

La aplicación de gestión de RedPelis maneja dos listas de información: una lista con la información de todos sus clientes y una lista con información de todas las películas disponibles. Estas listas, de tamaño variable, se cargan a partir de dos archivos de texto al abrir la aplicación, y se guardan en los mismos archivos cuando termina la ejecución.

La práctica consiste en desarrollar esta aplicación, en versión de consola, con la funcionalidad necesaria para gestionar la lista de clientes, la lista de películas y los visionados de películas por los clientes.

Para ello, crearemos un programa que empiece intentando cargar los contenidos iniciales de las listas a partir de los archivos `clientes.txt` y `pelis.txt`. En caso de que alguno de los archivos no exista, el programa empezará con una lista vacía. Después de cargar los archivos, se mostrará un menú con las siguientes opciones:

- 1.- Gestionar películas
- 2.- Gestionar clientes
- 3.- Gestión general
- 0.- Salir

1.1. Gestión de películas

La gestión de la lista de películas permite a los administradores del sistema dar de alta nuevas películas, modificar películas existentes o dar de baja películas que ya no están disponibles para alquiler. Esto se hace por medio del siguiente menú:

- 1.- Alta de película
- 2.- Datos de película
- 3.- Datos de todas las películas
- 4.- Edición de película
- 5.- Baja de película
- 0.- Volver al menú anterior

Las películas se guardan en una lista de tamaño variable, según la estructura vista en el temario (una estructura que contiene un array y un contador del número de posiciones ocupadas). El tamaño máximo de la lista es 30, pero tiene que ser sencillo cambiar el programa para admitir listas de mayor tamaño.

Para cada película, se guardará la siguiente información:

- **Código:** Un número entero positivo, único para cada película.
- **Título:** El título de la película, en una cadena de texto.
- **Género:** El género de la película; únicamente se contemplan los siguientes géneros: ciencia-ficción, fantasía, comedia, terror, drama, comedia-romántica.
- **Duración:** La duración de la película en minutos.
- **Precio:** El precio de cada visionado de la película (0 si es una descarga gratuita).
- **Disponible:** Indica si la película está disponible para ser visionada.

Las operaciones que se pueden realizar son:

- **Alta de una nueva película:** Comprueba que la lista de películas no esté llena. Si no lo está, pide al usuario los campos código (uno que no exista), título, género, duración y precio. Crea un nuevo registro para la película y lo guarda en la lista. Cuando se da de alta una película automáticamente se hace disponible. La lista debe mantenerse ordenada por código.
- **Datos de una película:** Pide al usuario un código de película. Si el código existe, muestra en la consola los datos de la película correspondiente.
- **Datos de todas las películas:** Muestra los datos completos de todas las películas de la lista. Se le da a elegir al usuario entre ver todas o ver sólo las disponibles.
- **Edición de una película:** Pide al usuario un código de película. Si el código existe, ofrece secuencialmente al usuario la posibilidad de modificar los campos título, género, duración y precio (el código no se puede modificar y para ponerla como no disponible hay que dar de baja la película).

- **Baja de una película:** Las películas no se pueden eliminar realmente, ya que es importante conservar los datos en el sistema. Cuando el usuario solicita dar de baja una película, lo que hacemos realmente es poner el campo “disponible” a falso. Esta opción pide al usuario un código y, si el código existe, marca la película indicada como no disponible.

1.2. Gestión de clientes

La gestión de clientes permite a los administradores del sistema dar de alta nuevos clientes, modificar sus datos y dar de baja a clientes. Esto se hace por medio del siguiente menú:

- 1.- Nuevo cliente
- 2.- Datos de cliente
- 3.- Datos de todos los clientes
- 4.- Editar cliente
- 5.- Baja de cliente
- 0.- Volver al menú anterior

Los clientes se guardan en otra lista de tamaño variable, de un tamaño máximo de 20, aunque será sencillo cambiar el programa para admitir listas de mayor tamaño.

Para cada cliente, se guardará la siguiente información:

- **NIF:** Una cadena de texto de 9 caracteres.
- **Nombre:** Una cadena de texto con el nombre del cliente.
- **Apellidos:** Una cadena de texto con los apellidos del cliente.
- **Género favorito:** Su género preferido, que tiene que ser uno de los géneros posibles en la lista de películas.
- **Películas visionadas:** Una lista, también de tamaño variable, que contiene un registro para cada película vista por ese cliente. Se pueden guardar un máximo de 40 películas visionadas. Cada registro incluye los siguientes campos:
 - Código de la película visionada.
 - Campo indicando si al cliente le gustó o no la película.

Las operaciones que se pueden realizar son:

- **Alta de un nuevo cliente:** Comprueba que la lista de clientes no esté llena. Si no lo está, pide al usuario los campos NIF (uno que no exista), nombre, apellidos y género favorito. La lista debe mantenerse ordenada por NIF. La lista de películas visionadas se crea vacía.

- **Datos de un cliente:** Pide al usuario un NIF. Si el NIF existe, muestra en la consola los datos detallados del cliente, incluyendo sus películas visionadas. Para cada película, se mostrará el título, el género y si le gustó al cliente o no.
- **Datos de todos los clientes:** Muestra los datos resumidos de todos los clientes. Para cada cliente se muestra el NIF, el nombre, los apellidos y el número total de películas visionadas.
- **Modificar los datos de un cliente:** Pide al usuario un NIF. Si el NIF existe, ofrece secuencialmente al usuario la posibilidad de modificar los campos nombre, apellidos y género favorito.
- **Baja de un cliente:** Debido a las exigencias de la LOPD, los datos de los clientes que se dan de baja no se pueden guardar en el sistema. Cuando el usuario solicita dar de baja a un cliente, el programa le pide el NIF y, si existe, elimina ese cliente de la lista. (Se pedirá confirmación, mostrando los datos del cliente que se va a dar de baja.)

1.3. Gestión general

La gestión general permite a los administradores del sistema realizar distintas operaciones con las listas de clientes y películas:

- 1.- Nuevo visionado
- 2.- Exportar lista de clientes
- 3.- Estadísticas de películas
- 0.- Volver al menú anterior

Las operaciones se describen en detalle a continuación:

- **Nuevo visionado:** Pide al usuario un código de película, un NIF de cliente y si al cliente le gustó la película. En caso de que tanto la película como el cliente existan, añade un nuevo registro al final de la lista de películas visionadas del cliente.
- **Exportar la lista de clientes:** Pide al usuario un nombre de archivo de texto y escribe en ese archivo una lista con los datos básicos de los clientes (apellidos, nombre, NIF y número de películas visionadas) ordenada por apellidos. La operación no debe modificar la lista original.
- **Estadísticas de las películas:** Muestra en la consola un listado de todas las películas, indicando para cada película el título, el número de veces que ha sido visionada y el número de veces en los que la película le gustó al cliente.

2. Guía de implementación

A la hora de completar la práctica, deberás seguir las siguientes indicaciones.

2.1. Organización en módulos

La práctica deberá organizarse en módulos, con un módulo independiente para cada tipo principal. Se propone la creación de los siguientes módulos:

Género

Declara un tipo de datos `tGenero`, un enumerado para gestionar los géneros. También ofrece las siguientes operaciones:

- `leerGenero()`: Pide un género al usuario y devuelve una variable de tipo `tGenero` con el género leído.
- `escribirGenero()`: Dada una variable de tipo `tGenero`, escribe el nombre del género en la consola.

Película

Declara un tipo de datos `tPelícula` para guardar la información de una película y las siguientes operaciones:

- `leerPelícula()`: No recibe ningún valor de entrada. Pide al usuario los datos necesarios para la ficha de una película y devuelve como salida una estructura `tPelícula` debidamente inicializada.
- `mostrarPelícula()`: Recibe una película y escribe todos sus datos en la consola, debidamente formateados.
- `editarPelícula()`: Recibe una película y ofrece al usuario la posibilidad de editar los campos título, género, duración y precio.

Lista de películas

Declara un tipo de datos `tListaPelis` para guardar la lista de películas, así como las siguientes operaciones:

- `listaPelículasLlena()`: Función que recibe como parámetro una lista de películas y devuelve un booleano indicando si esa lista está llena o no.
- `buscarPelícula()`: Recibe como parámetro de entrada una lista de películas y un código. Devuelve un booleano indicando si se ha encontrado o no y, en caso afirmativo, la posición en la que se encuentra la película.
- `insertarPelícula()`: Recibe una variable de tipo `tPelícula` y una lista de películas. Inserta la película en la lista en la posición que corresponda por código y devuelve la lista actualizada.

- **bajaPelicula()**: Recibe el código de una película y una lista de películas. Busca la película en la lista y, si está, la marca como no disponible. Devuelve un booleano indicando si se ha podido realizar la operación o no.

Cliente

Declara un tipo de datos **tCliente** y las siguientes operaciones:

- **leerCliente()**: No recibe ningún parámetro de entrada. Lee los datos de un cliente, inicializa su lista de visionados como una lista vacía y devuelve como salida una estructura **tCliente** debidamente creada. Se deberá comprobar que el NIF es correcto (8 dígitos y una letra al final, que se convertirá a mayúscula).
- **mostrarCliente()**: Recibe como entrada un **tCliente**, un booleano **mostrarDetalles** y la lista completa de películas. Si **mostrarDetalles** es falso, muestra por pantalla los datos del cliente, incluyendo el número de películas que ha visionado. Si **mostrarDetalles** es cierto, se mostrarán también los datos detallados de las películas visionadas tal y como se indica en la funcionalidad de *mostrar los datos de un cliente* (título, género y si la película le ha gustado al cliente o no).
- **editarCliente()**: Recibe como entrada un **tCliente** y ofrece al usuario la posibilidad de editar los campos nombre, apellidos y género favorito.
- **altaVisionado()**: Recibe como entrada un **tCliente** y los datos de un visionado (el código de la película y si le gustó o no). Devuelve como salida la estructura **tCliente** debidamente actualizada.
- **datosVisionado()**: Recibe como entrada un **tCliente** y el código de una película. Devuelve dos enteros: uno indicando cuántas veces ha visionado la película (podrían ser más de una) y cuántas de esas veces le gustó la película.

Lista de clientes

Declara un tipo de datos **tListaClientes** para guardar la lista de clientes, así como las siguientes operaciones:

- **listaClientesLlena()**: Función que recibe como parámetro una lista de clientes y devuelve un booleano indicando si la lista está llena o no.
- **buscarCliente()**: Recibe como parámetro de entrada una lista de clientes y un NIF. Devuelve un booleano indicando si se ha encontrado el cliente o no y, en caso afirmativo, la posición en la que se encuentra el cliente.

- `insertarCliente()`: Recibe una variable de tipo `tCliente` y una lista de clientes. Inserta el cliente en la lista en la posición que corresponda por NIF y devuelve la lista actualizada.
- `bajaCliente()`: Recibe un NIF de cliente y una lista de clientes. Busca al cliente en la lista y lo elimina completamente.

2.2. Formato de los archivos de texto

`clientes.txt`

Datos de los clientes. Por cada cliente una serie de líneas, primero con sus datos y luego con las películas visionadas:

NIF ("X" para terminar) ↵

Nombre ↵

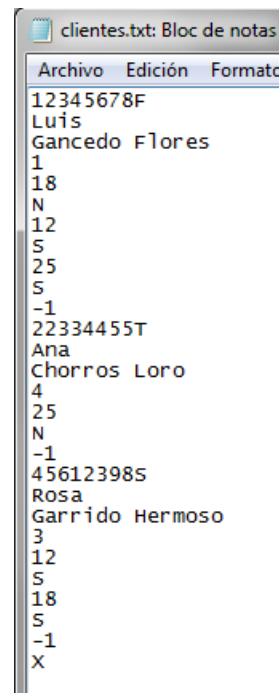
Apellidos ↵

Género preferido (entero con el código del enumerado: 0, 1, 2, 3, 4 ó 5) ↵

Por cada película visionada:

Código de la película (-1 para terminar) ↵

'S' o 'N' (le gustó o no la película) ↵



```

12345678F
Luis
Gancedo Flores
1
18
N
12
S
25
S
-1
22334455T
Ana
Chorros Loro
4
25
N
-1
45612398S
Rosa
Garrido Hermoso
3
12
S
18
S
-1
X
  
```

`pelis.txt`

Datos de las películas. Por cada película una serie de líneas:

Código de la película (-1 para terminar) ↵

Título ↵

Género (entero con el código del enumerado: 0, 1, 2, 3, 4 ó 5) ↵

Duración (entero) ↵

Precio (real) ↵

'S' o 'N' (disponible o no la película) ↵



```

6
Avatar
1
123
3.95
N
12
Burried
3
98
2.95
S
18
2012
1
115
3.95
S
22
Extraterrestre
2
92
2.95
S
25
La invencion de Hugo
1
132
4.25
S
-1
  
```

2.3. Otras indicaciones

- Las listas de tamaño variable deberán implementarse con la estructura de array + contador vista en el Tema 6.
- Siempre que se pueda, las operaciones de búsqueda serán binarias.
- Todos los módulos se implementarán con archivos `.h` y `.cpp`. En el archivo `.h` se incluirán los prototipos de las funciones públicas con comentarios describiendo qué hace cada función, qué datos recibe y qué datos devuelve.
- Los módulos `tListaPeliculas` y `tListaClientes` únicamente realizan operaciones con listas, y nunca deben escribir nada en pantalla. Pueden devolver valores al programa que los llame para que éste escriba en pantalla lo que sea necesario.

3. Entrega de la práctica

La práctica se entregará a través del Campus Virtual. Se habilitará una nueva tarea **Entrega de la Práctica 4** que permitirá subir un único archivo. Hay que subir un archivo comprimido `GrupoXX.zip` que contenga los archivos `.h` y `.cpp` del proyecto (y ninguno más).

Fin del plazo de entrega: **10 de mayo a las 23:55.**