



# Universidad Europea

**UNIVERSIDAD EUROPEA DE MADRID**

**ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO**

**GRADO EN INGENIERIA INFORMÁTICA**

**ACTIVIDAD COLABORATIVA 3**

**Programación con estructuras lineares**

**Ing. Jorge García**

**EJERCICIOS PRACTICOS DE GESTIÓN DE MEMORIA**

**Grupo 1:**

**MIGUEL SÁNCHEZ BERNUY**

**GERÓNIMO BASSO**

**JORGE CASTILLA COELLO**

**VICTOR MARTÍN LORENZO**

**DIOGO RIBEIRO da SILVA**

**CURSO 2022-2023**

# 1 RESUMEN

---

## 1.1 TABLA RESUMEN

	DATOS
<b>Nombre y apellidos:</b>	MIGUEL SÁNCHEZ BERNUY GERÓNIMO BASSO JORGE CASTILLA COELLO VICTOR MARTÍN LORENZO DIOGO RIBEIRO DA SILVA
<b>Título del proyecto:</b>	Ejercicios prácticos de gestión de memoria
<b>Directores del proyecto:</b>	Jorge García
<b>El proyecto se ha realizado en colaboración de una empresa o a petición de una empresa:</b>	NO
<b>El proyecto ha implementado un producto:</b> (esta entrada se puede marcar junto a la siguiente)	NO
<b>El proyecto ha consistido en el desarrollo de una investigación o innovación:</b> (esta entrada se puede marcar junto a la anterior)	NO
<b>Objetivo general del proyecto:</b>	Demostrar que somos capaces de completar las tareas aplicando aptitudes adquiridas

## 2 ÍNDICE

---

1	Resumen.....	2
1.1	TABLA RESUMEN .....	2
2	Índice .....	3
3	Índice de Figuras.....	4
4	Resumen del proyecto.....	5
4.1	Contexto y justificación .....	5
4.2	Planteamiento del problema .....	5
4.3	Objetivos del proyecto .....	5
5	Guía de usuario.....	6
5.1	Introducción a la guía de usuario .....	6
5.2	Ejercicio 1.....	6
5.3	Ejercicio 2.....	7
5.4	Ejercicio 3 .....	14
6	Objetivos.....	17
6.1	Objetivos generales .....	17
6.2	Objetivos específicos .....	17
6.3	Beneficios del proyecto .....	17
7	Clases y métodos .....	18
7.1	Ejercicio 1.....	18
7.2	Ejercicio 2.....	19
7.3	Ejercicio 3.....	20
8	Enlaces y Referencias .....	23
9	Conclusiones.....	24
9.1	Conclusiones del trabajo .....	24

### 3 ÍNDICE DE FIGURAS

---

Ilustración 1: Manual de Usuario 1.1.....	6
Ilustración 2: Manual de Usuario 2.1.....	7
Ilustración 3: Manual de Usuario 2.2.....	8
Ilustración 4: Manual de Usuario 2.3.....	9
Ilustración 5: Manual de Usuario 2.4.....	10
Ilustración 6: Manual de Usuario 2.5.....	11
Ilustración 7: Manual de Usuario 2.6.....	11
Ilustración 8: Manual de Usuario 2.7.....	12
Ilustración 9: Manual de Usuario 2.8.....	12
Ilustración 10: Manual de Usuario 2.9.....	13
Ilustración 11: Manual de Usuario 2.10.....	13
Ilustración 12: Manual de Usuario 3.1.....	14
Ilustración 13: Manual de Usuario 3.2.....	15
Ilustración 14: Manual de Usuario 3.3.....	15
Ilustración 15: Manual de Usuario 3.4.....	15
Ilustración 16: Manual de Usuario 3.5.....	16
Ilustración 17: Manual de Usuario 3.6.....	16

## **4 RESUMEN DEL PROYECTO**

---

### **4.1 CONTEXTO Y JUSTIFICACIÓN**

Varias empresas nos han comunicado para mejorarles sus sistemas de almacenamiento de datos ya que están algo obsoletos respecto a los tiempos que corren hoy en día.

### **4.2 PLANTEAMIENTO DEL PROBLEMA**

Para resolver dicha necesidad hemos decidido desarrollar un sistema de almacenamiento en C++ mediante la aplicación de vectores. Los mismos se encargarán de almacenar y distribuir toda la información necesaria al programa para que funcione adecuadamente.

### **4.3 OBJETIVOS DEL PROYECTO**

Como objetivo del proyecto hemos decidido establecernos la victoria una vez el cliente esté completamente feliz con el programa desarrollado para su empresa. Una vez nos confirme que es completamente funcional y lo que buscaba daremos por finiquitado el trabajo.

## 5 GUÍA DE USUARIO

### 5.1 INTRODUCCIÓN A LA GUÍA DE USUARIO

Este manual de usuario es un breve documento que explica las diferentes funciones del código. A través de él, podrás aprender a navegar y utilizar el programa de una manera intuitiva y agradable. Con un diseño visual, podrás comprender fácilmente las distintas acciones que el programa te ofrece. ¡Con esta guía, podrás sacarle el máximo partido a tu software!

### 5.2 EJERCICIO 1

Con este ejercicio presentamos una gran innovación respecto a los anteriores, por primera vez contamos con una consola operativa en vez de un menú de opciones

```
"/Users/jorge/Documents/Universidad/Año Académico 22:23/Ccode/proyecto-1-discografia/cmake-build-debug/proyecto_1_discografia"
Introduzca la funcion con los atributos separados por espacio: add Storyteller Carrie_Underwood Country
Introduzca la funcion con los atributos separados por espacio: nuevo Blown_Away Carrie_Underwood Country
Introduzca la funcion con los atributos separados por espacio: new New_Creation Mac_Powell Cristiana
Introduzca la funcion con los atributos separados por espacio: crear Morningtide Celtic_Worship Cristiana
Introduzca la funcion con los atributos separados por espacio: buscar nombre Storyteller
ID: 1, Nombre: Storyteller, Grupo: Carrie_Underwood, Genero: Country
Introduzca la funcion con los atributos separados por espacio: seach name storyteler
Introduzca una funcion correcta
Introduzca la funcion con los atributos separados por espacio: search name storyteler
ID: 1, Nombre: Storyteller, Grupo: Carrie_Underwood, Genero: Country
Introduzca la funcion con los atributos separados por espacio: srch genero cristiana
ID: 3, Nombre: New_Creation, Grupo: Mac_Powell, Genero: Cristiana
ID: 4, Nombre: Morningtide, Grupo: Celtic_Worship, Genero: Cristiana
Introduzca la funcion con los atributos separados por espacio:
```

*Ilustración 1: Manual de Usuario 1.1*

Los comandos, y su estructura es la siguiente

Para crear un nuevo álbum: {(Comando) (Nombre) (Grupo) (Genero)}

Los comandos para crear un nuevo álbum son: add, new, nuevo, añadir, create, crear.

Por ejemplo, el siguiente es un comando valido: add Storyteller Carrie\_Underwood Country

Para buscar un álbum en el sistema: {(Comando) (Por) (A buscar)}

Los comandos para buscar son: buscar, search, srch.

Se puede buscar por nombre, grupo o genero con los siguientes elementos por:

nombre	grupo	genero
name	cantante	estilo
	singer	style
	group	

	banda	
	band	

Por ejemplo, el siguiente es un comando valido: buscar nombre Storyteller

ID: 1, Nombre: Storyteller, Grupo: Carrie\_Underwood, Genero: Country

Gracias a otra innovación, es posible buscar por similitud, es decir, puede equivocarse en algún elemento a la hora de introducir el nombre y debería de funcionar, por otra parte, los comandos y sus atributos deben de escribirse de forma precisa.

Para salir puede utilizar: exit, salir, end

### 5.3 EJERCICIO 2

El programa comienza mostrando un menú con varias opciones, donde el usuario puede elegir ingresando la opción del menú que quiera ejecutar. El menú es el siguiente:

```
---- Bienvenido a nuestro VIDEOCLUB, seleccione una de las opciones de nuestro menu-----  
Opciones Menu:  
1. Añadir Disco  
2. Mostar listado de todos los discos  
3. Buscar discos por titulo  
4. Mostrar libros disponibles para alquiler  
5. Alquilar disco  
6. Devolver disco  
7. Terminar programa
```

*Ilustración 2: Manual de Usuario 2.1*

En este caso, el usuario selecciona la opción “1”. El programa pide varios datos para poder añadir un disco al listado de discos. Por lo tanto, se muestra lo siguiente:

```
1
Ingrese el titulo:
Lord of the rings
Ingrese el precio:
9.99
El disco esta en formato DVD? (s/n) :
s
Opciones Menu:
1. Añadir Disco
2. Mostar listado de todos los discos
3. Buscar discos por titulo
4. Mostrar libros disponibles para alquilar
5. Alquilar disco
6. Devolver disco
7. Terminar programa
```

*Ilustración 3: Manual de Usuario 2.2*

El usuario ingreso el título "Lord of the rings", junto con el precio "9.99" y la letra "s" para referirse que el disco está en formato DVD. Una vez ingresada toda esta información sobre el nuevo disco, junto con los chequeos necesarios para que ingresen la información correctamente, volvemos al menú y seguimos con el flujo. El usuario escribe "2", a lo que el programa procede a mostrar un listado de todos los discos que hay registrados en el sistema, junto con la información que hay sobre ellos, se muestra de la siguiente forma:



2

```
*Titulo: Star wars: the phantom menance
Precio: 9.990000
En DVD: Si
En BLR: No
¿Esta alquilado? No
*Titulo: Star wars: attack of the clones
Precio: 12.990000
En DVD: No
En BLR: Si
¿Esta alquilado? No
*Titulo: Star wars: revange of the sith
Precio: 7.990000
En DVD: Si
En BLR: No
¿Esta alquilado? No
*Titulo: Star wars: a new hope
Precio: 9.990000
En DVD: No
En BLR: Si
¿Esta alquilado? No
*Titulo: Star wars: the empire strikes back
Precio: 8.990000
En DVD: Si
En BLR: No
¿Esta alquilado? No
```

*Ilustración 4: Manual de Usuario 2.3*

```
*Titulo: Star wars: return of the jedi
Precio: 13.990000
En DVD: No
En BLR: Si
¿Esta alquilado? No
*Titulo: Star wars: the force awakens
Precio: 12.990000
En DVD: Si
En BLR: No
¿Esta alquilado? No
*Titulo: Rogue one: a star wars story
Precio: 6.990000
En DVD: No
En BLR: Si
¿Esta alquilado? No
*Titulo: Star wars: the las jedi
Precio: 9.990000
En DVD: Si
En BLR: No
¿Esta alquilado? No
*Titulo: Solo: a star wars story
Precio: 15.990000
En DVD: No
En BLR: Si
¿Esta alquilado? No
*Titulo: Star wars: the rise of skywalker
Precio: 16.990000
En DVD: Si
En BLR: No
¿Esta alquilado? No
*Titulo: Lord of the rings
Precio: 9.990000
En DVD: Si
En BLR: No
¿Esta alquilado? No
```

*Ilustración 5: Manual de Usuario 2.4*

Continuando con el programa, volvemos a mostrar el menú principal. A lo que el usuario selecciona "3" para buscar el disco por un título.

```
Opciones Menu:  
1. Añadir Disco  
2. Mostar listado de todos los discos  
3. Buscar discos por titulo  
4. Mostrar libros disponibles para alquiler  
5. Alquilar disco  
6. Devolver disco  
7. Terminar programa
```

*Ilustración 6: Manual de Usuario 2.5*

Luego de ingresar "3" el usuario ingresa el nombre del título que va a buscar. En este caso "Lord of the rings". A lo que se devuelve toda la información sobre ese título. Con el siguiente formato:

```
3  
Ingrese el titulo:  
Lord of the rings  
*Titulo: Lord of the rings  
Precio: 9.990000  
En DVD: Si  
En BLR: No  
¿Esta alquilado? No
```

*Ilustración 7: Manual de Usuario 2.6*

Continuando, volvemos a desplegar el menú principal y elegimos la opción 4.

```
Opciones Menu:
1. Añadir Disco
2. Mostar listado de todos los discos
3. Buscar discos por titulo
4. Mostrar libros disponibles para alquiler
5. Alquilar disco
6. Devolver disco
7. Terminar programa
4
Listado de discos disponibles para alquiler:
```

*Ilustración 8: Manual de Usuario 2.7*

En este caso, como no se han reservado ningún disco se muestra el mismo menú que la opción 2. En el caso de que algún disco este alquilado, en este menú no aparecerá dicho disco, pero si los que no estén alquilados.

Volvemos a mostrar el menú y seleccionamos la opción 5. A lo que escribimos el título del disco que queremos alquilar, en este caso *"Lord of the rings"*. Si el disco no está disponible, no se podrá realizar esta acción. Se muestra lo siguiente:

```
Opciones Menu:
1. Añadir Disco
2. Mostar listado de todos los discos
3. Buscar discos por titulo
4. Mostrar libros disponibles para alquiler
5. Alquilar disco
6. Devolver disco
7. Terminar programa
5
Introduzca el nombre del disco que quiere alquilar:
Lord of the rings
Se ha alquilado con exito el libro
```

*Ilustración 9: Manual de Usuario 2.8*

Luego, volvemos al menú principal, y seleccionamos la opción “6”, para devolver el disco que habíamos alquilado previamente, u otro disco que hayamos reservado previamente. Ingresamos el título del disco, y si estaba alquilado imprimiremos el mensaje “Se ha devuelto el libro con éxito”.

```
Opciones Menu:
1. Añadir Disco
2. Mostar listado de todos los discos
3. Buscar discos por titulo
4. Mostrar libros disponibles para alquiler
5. Alquilar disco
6. Devolver disco
7. Terminar programa
6
Introduzca el nombre del disco que quiere devolver:
Lord of the rings
Se ha devuelto el libro con exito
```

*Ilustración 10: Manual de Usuario 2.9*

Por último, seleccionamos la opción 7 para dejar de correr el programa.

```
Se ha devuelto el libro con exito
Opciones Menu:
1. Añadir Disco
2. Mostar listado de todos los discos
3. Buscar discos por titulo
4. Mostrar libros disponibles para alquiler
5. Alquilar disco
6. Devolver disco
7. Terminar programa
7
```

*Ilustración 11: Manual de Usuario 2.10*

## 5.4 EJERCICIO 3

### 5.4.1 CONTEXTUALIZACIÓN

¡Bienvenidos a nuestro programa de encuestas sobre el color de lightsaber!

Si eres fanático de Star Wars, seguramente te has preguntado cuál sería tu color de lightsaber si tuvieras la habilidad de usar la Fuerza. ¡Ahora puedes averiguarlo con nuestro programa!

Nuestro programa te hará una serie de preguntas para conocer tus gustos y personalidad, y con base en tus respuestas, te asignará un color de lightsaber que se relaciona con tu forma de ser. ¿Serás un maestro Jedi con un lightsaber verde, o un guerrero Sith con uno rojo?  
¡Descúbrelo ahora!

Además, nuestro programa te ofrecerá información interesante sobre el significado de cada color de lightsaber y cómo se relaciona con la personalidad y los gustos de las personas. ¡Es una forma divertida de conocerte mejor y de compartir tus resultados con tus amigos!

¡Anímate a participar en nuestra encuesta y descubre tu verdadero color de lightsaber!

### 5.4.2 GUIA DE USUARIO

Tras inicializar el programa se abre un menú de 4 opciones

```
SE RECOMIENDA USAR PANTALLA COMPLETA PARA MEJOR EXPERIENCIA
1- Insertar datos del Padawan
2- Ver datos de todos los Padawans
3- Filtrar datos Padawans
4- Salir del Programa

Introduce la opcion que deseas realizar:
```

*Ilustración 12: Manual de Usuario 3.1*

En el caso 1 se inicializa el test en el que deberás introducir un nombre y contestar las preguntas del test para saber qué tipo de lightsaber te corresponde.

```
Escriba su nombre, joven padawan: Victor
1- Cruel
2- Agresivo
3- Protector
4- Pacifico
5- Acrobatico
6- Balanceado

cual describe tu estilo de Lucha?:
```

Ilustración 13: Manual de Usuario 3.2

Tras terminar el test puedes saber el tipo de lightsaber del Jedi y su porcentaje de este

```
Porcentaje de Jedi = 16%

El color del light saber es: naranja
```

Ilustración 14: Manual de Usuario 3.3

Después de hacer el test se añaden los datos al vector para más tarde buscar información

Si eliges la opción 2 sale toda la lista y si eliges la opción 3 puedes buscar a un usuario en concreto los datos del padawan salen acompañados de una breve descripción del color de su sable.

Opción 2:

```
Introduce la opcion que deseas realizar:
Padawan 1 | : Victor
Color Light Saber : rojo
-- BREVE DESCRIPCION --
Los sables de luz rojos a menudo se asocian con los Sith o los Jedi oscuros, y simbolizan la ira y la determinacion.
El código Sith predica que la ira, el odio y la agresion son la forma de lograr
los objetivos de uno, lo que probablemente explica por que el rojo se convirtio
en el color elegido por muchos Lores Sith

Padawan 2 | : Gero
Color Light Saber : azul
-- BREVE DESCRIPCION --
Tal como el verde es uno de los mas comunes
Significa justicia, verdad y serenidad.

Padawan 3 | : Jorge
Color Light Saber : amarillo
-- BREVE DESCRIPCION --
Los sables de luz amarillos tambien son relativamente raros,
pero representan el poder del sol y la esperanza para el futuro.
En Star Wars: The Clone Wars, Ahsoka Tano empuñó un sable de luz amarillo
para simbolizar su nuevo papel como lider y protectora.
```

Ilustración 15: Manual de Usuario 3.4

### Opción 3:

```
Introduce el nombre del padawan:Victor
Padawan 1: Victor
Color Light Saber: rojo
-- BREVE DESCRIPCION --
Los sables de luz rojos a menudo se asocian con los Sith o los Jedi oscuros, y simbolizan la ira y la determinacion.
El código Sith predica que la ira, el odio y la agresion son la forma de lograr
los objetivos de uno, lo que probablemente explica por qué el rojo se convirtió
en el color elegido por muchos Lores Sith
```

*Ilustración 16: Manual de Usuario 3.5*

Con la opción 4 sales del programa.

```
Que la fuerza te acompañe.

Process finished with exit code 0
```

*Ilustración 17: Manual de Usuario 3.6*



## 6 OBJETIVOS

---

### 6.1 OBJETIVOS GENERALES

Tarea 1 y 2: El objetivo general de dicho proyecto es satisfacer las necesidades de los clientes en su totalidad y mejorarles sus sistemas anticuados de contabilización.

Tarea 3: Ofrecer una experiencia única y divertida a nuestros usuarios. Mediante la cual podrán presumir de sus resultados con los amigos.

### 6.2 OBJETIVOS ESPECÍFICOS

Tarea 1: Implementar en C++ un programa que sea capaz de realizar una búsqueda eficiente que permita a los usuarios encontrar rápidamente los álbumes que buscan, ya sea por título, grupo o género.

Tarea 2: Implementar en C++ un programa que sea capaz de mejorar la experiencia del usuario al interactuar con el sistema de alquileres del videoclub.

Tarea 3: Ofrecer una herramienta divertida y educativa que permita a las personas conocerse mejor mediante la implementación de una encuesta en C++.

### 6.3 BENEFICIOS DEL PROYECTO

Afianzar términos, conceptos y técnicas aprendidos a lo largo del curso. Además de servir como repaso y practica de la parte práctica de la asignatura.

Por último, facilitar la creación de otros posibles códigos mediante la reutilización de los ejercicios propuestos.

## 7 CLASES Y MÉTODOS

---

### 7.1 EJERCICIO 1

Dentro del proyecto “proyecto-1-discografia” tenemos 5 archivos principales, siendo estos “main.cpp”, “Album.h”, “consola.h”, “FuncionesDistancia.h” y “VectorPEL.h”. Cada uno de estos archivos cumple funciones distintas, las cuales vamos a proceder a enumerar:

#### 7.1.1 Clase Main

La clase main, unicamente llama al constructor de consola

#### 7.1.2 Clase menu

La clase menu, contiene 4 vectores, uno de punteros a Albums que contiene los albums y los otros 3 contienen respectivamente los nombres, grupos y generos de los album, está hecho de esta manera para poder usar las funciones de similitud, todo esto se encuentra contenido en el constructor.

#### 7.1.3 Clase FuncionesDistancia

La clase FuncionesDistancia, no contiene ningún atributo, pero si contiene 2 métodos llamados similitudExiste y masParecido, el primero devuelve true si la distancia entre el más parecido y el introducido es menor de 4, el otro, devuelve el más parecido.

El funcionamiento de estos métodos, lo procedo a detallar, dado que ambos funcionan igual hasta lo que devuelven, procederé a comentarlos conjuntamente, primero se introduce un string y un puntero a un vector de strings. A continuación, se sitúa un entero al número máximo que permite, para hacer de infinito, la distancia máxima.

Por cada elemento dentro del vector, hacemos una comparación carácter por carácter, si las letras no coinciden, aumentamos en uno la distancia, si la distancia actual es menor que la mínima que hemos calculado se convierte en la mínima y situamos ese string como el más parecido

#### 7.1.4 Clase Album

La clase Álbum, contiene 3 strings, nombre, grupo y género, un constructor vacío y un constructor por parámetros, también contiene los getters necesarios y un método toString que devuelve un string formateado listo para imprimir por consola.

#### 7.1.5 Clase VectorPEL

Esta clase fue propocionada por el profesor Jorge Garcia, la hicimos en clase con el objetivo de poder simular un Vector creado por nosotros, trabajando con los punteros.

## 7.2 EJERCICIO 2

Dentro del proyecto “*proyecto-2-videoClub*” tenemos 3 archivos principales, siendo estos “*main.cpp*”, “*Disco.h*” y “*VectorPEL.h*”. Cada uno de estos archivos cumplen funciones distintas, las cuales vamos a proceder a explicar individualmente ahora:

### 7.2.1 Clase Main:

En la clase main se encuentra el código que vamos a ejecutar, dentro de este tenemos tres funciones.

- Funciones:
  - void inicializarDiscos(VectorPEL<Disco> &discos)
  - void cinearlo()
  - int main()

En la función **inicializarDiscos** lo que hago es creo varios discos y los añado al vector para que el programa tenga algunos discos inicializados desde el principio, y el vector no este vacío.

Es la función **cinearlo** lo que hago es limpiar el cin para evitar problemas al pasarle datos por terminal

En la función **int main()** tenemos la mayor parte de código. Se empieza por un while donde pedimos que nos ingresen la opción para el menú principal, dependiendo de esta opción entramos a un caso en el cual se realiza lo que se pidió por menú previamente. Los casos disponibles son:

1. Añadir Disco: Menu para agregar un disco al programa
2. Mostar listado de todos los discos: Se imprimen todos los discos en el programa
3. Buscar discos por título: Menu para buscar discos por titulo
4. Mostrar libros disponibles para alquiler: Se devuelven todos los libros que no esten alquilados
5. Alquilar disco: Se ingresa un título y setea como alquilado el libro, en caso de que el libro ya este alquilado, nose permite esta acción.
6. Devolver disco: Se ingresa un título y setea como disponible el libro.
7. Terminar programa: Se termina el loop principal y se termina de ejecutar el programa

### 7.2.2 Clase Disco:

La clase disco se utiliza para crear un modelado del objeto “*Disco*”, esta clase contiene los atributos que conforman a un disco, su constructor, varios getters y setters para poder acceder

a su información y un método toString para poder imprimir la información dentro del objeto "Disco".

### 7.2.3 Clase VectorPEL:

Esta clase fue proporcionada por el profesor Jorge Garcia, la hicimos en clase con el objetivo de poder simular un Vector creado por nosotros, trabajando con los punteros.

## 7.3 EJERCICIO 3

### 1. Clase LightSaber:

- Variables de la clase: username de tipo string: usada para almacenar el nombre del usuario colorLight de tipo string: usada para almacenar el color de lightsaber del usuario.
- Constructor de la clase LightSaber con sus respectivas variables
- Constructor vacío LightSaber
- Método mostrar que enseña los datos de la clase LightSaber

### 2. Método Cinearlo:

- Método de tipo void cinearlo: Sirve para reiniciar y limpiar los datos cin.

### 3. Método detectorLightS:

- Método de tipo string que contiene un quiz para identificar el color del light Saber del usuario que será lo que retornará.

*Creación de las variables:*

- variables de tipo int (n1,n2,n3,n4,n5,n6) -> serán usadas para almacenar cada una de las respuestas \* variables p1 = usada para calcular el nivel de jedi que determinan el color del light saber
- Variable de tipo array donde almacenamos las respuestas int listaN[]
- cout que muestra por pantalla las diferentes respuestas validas que determinaran el color del light saber
- cin donde se almacena la respuesta
- invocación del método cinearlo
  
- for que recoge el array listaN donde manejamos las respuestas del usuario \* y dependiendo de la respuesta le sumamos a la variable int p1 que determina \* el

nivel de sith, cuanto más bajo sea el número de dicha variable más cerca \* del lado oscuro está el usuario.

- if donde analizamos el tamaño de la variable p1 para ver \* el nivel de oscuridad del usuario y en base a dicho número retornamos un \* color del light saber

#### **4. Método descLightSaber:**

- Método de tipo string que pide otro string por parámetro \* dependiendo del string que se pasa por parámetro, en este caso será el color \* del ligh saber, el método retornará una descripción adecuada a dicho color.

#### **5. Método main:**

- Método tipo int main encargado de la ejecución del programa
- Declaración de VectorPEL destinado a la clase LigthSaber con referencia "starWars"
- Variable de tipo int de nombre caso que se usara para determinar el caso que el usuario quiera usar
- While con la variable bool continuar como parámetro \* este while se encarga de gestionar toda la ejecución del usuario
- cout que indica las diversas opciones que tiene el usuario \* para introducir
- sincronizamos y limpiamos la cache de los cin
- pedimos que el usuario introduzca uno de los casos mostrados por pantalla y lo almacenamos en la variable int caso
- variable donde almacenamos el tamaño del vector

#### *Switch:*

1. inicialización de la string user que será almacenada en un cin que se pedirá por pantalla, esta variable incluye el nombre del padwan/usuario String color que será determinado por el método detectoLightS Método que almacena los datos de registro del usuario en el vector
2. Verificamos si existen datos introducidos en el vector con la variable num. For que recoge todos los elementos del vector. Asignamos una referencia a la clase lightsaber a los datos que estamos recogiendo del vector. A través de la variable de la clase que tiene ahora los datos del vector podemos filtrar por nombre y usuario, mostramos ambos por pantalla identificado su posición en el vector a través del int i perteneciente al for. Para tener la descripción del color invocamos al método descLightSaber y le pasamos su respectivo color por parámetro En caso de que no haya ningún dato almacenado se mostrara dicho mensaje por pantalla.
3. Caso bastante idéntico al 2º pero filtrando los datos por usuario. A través de la referencia tipo LightSaber nombrada nombre, verificamos si el nombre

introducido por usuario esta almacenado en el vector, en caso de que si lo mostramos por pantalla.

4. Asignamos la variable continuar false para terminar el while que gestiona la ejecución del programa
5. Default: función perteneciente al método switch que se ejecuta cuando el caso introducido no existe

### **7.3.1 Clase VectorPEL**

Esta clase fue proporcionada por el profesor Jorge Garcia, la hicimos en clase con el objetivo de poder simular un Vector creado por nosotros, trabajando con los punteros.

## 8 ENLACES Y REFERENCIAS

---

En este apartado, presentamos los enlaces a todos los repositorios de GitHub creados, se crearon 6 repositorios en total, uno para cada ejercicio. Los repositorios son de tipo público, por lo cual cualquier puede acceder.

- Ejercicio 1:
  - <https://github.com/Geronimo-Basso/proyecto-1-discografia>
- Ejercicio 2:
  - <https://github.com/Geronimo-Basso/proyecto-2-videoclub>
- Ejercicio 3:
  - <https://github.com/Geronimo-Basso/proyecto-3-extra>

## 9 CONCLUSIONES

---

### 9.1 CONCLUSIONES DEL TRABAJO

Después de completar las actividades en las páginas anteriores, podemos concluir que la actividad grupal 3 ha sido finalizada con éxito. Como equipo, valoramos altamente esta actividad porque nos permitió mejorar nuestras habilidades en temas como vectores y el manejo de C++ en general. Además, continuamos desarrollando nuestra destreza en el uso de herramientas de desarrollo como GitHub y Git.

Estamos seguros de que nuestro proyecto cumplirá con las expectativas tanto de los clientes que adquirirán el software como de los “guerreros de la galaxia” que accederán a realizar la encuesta sobre su color de lightsaber.