

# Implementación de las opciones que faltan

Sigues adelante con las opciones que tiene el `switch` para manejar el menú que muestras al usuario. La opción a desarrollar ahora es la que termina una transmisión.

**TRMINA\_TRNSMSN** : Esta opción, como su nombre lo indica, debe terminar (o cerrar) una transmisión que esté abierta. El orden en que debes realizar las distintas acciones es el siguiente:

1. Mostrarle al usuario los discos que tienen transmisiones activas usando el método que programaste para ello. Como este método te regresa una cadena, la llamada la colocas dentro de un `println`.

```
610     case TRMINA_TRNSMSN:
611         /* primero se localiza la transmision */
612         System.out.println("Elige el disco cuya transmision deseas "
613                             + "terminar");
614         System.out.println(mstraActivos("Discos con transmisiones"
615                                         + " abiertas"));
```

2. Usando el método `pideNum`, le pides al usuario que digite la posición del disco del que desea terminar una transmisión, guardándolo en el entero `cualDisco`, que declaraste al principio del método.

```
616         cualDisco = pideNum(cons, "Elige el numero de disco, ",
617                               0, numDiscos - 1);
```

3. Si el disco elegido por el usuario no existe, das un mensaje y sales de la condicional.

```
618         if ( cualDisco == -1 ) { // Disco incorrecto
619             System.out.println("El disco " + cualDisco + " no existe");
620             break;
621         }
```

4. Si el disco elegido no tiene transmisiones activas sales del método, avisándole al usuario que el disco elegido no tiene transmisiones sin terminar.

```
618         if (catalogo[cualDisco].getActivas() <= 0) {
619             System.out.println("El disco [" + cualDisco + "] "
620                                 + "no tiene transmisiones activas");
621             break;
622         }
```

5. Si llegas acá es porque no has salido de la condicional por algún error en los datos del usuario. Usas el método del catálogo que termina una transmisión de un disco dado y verificas si se pudo terminar la transmisión. Das al usuario el mensaje correspondiente y sales del `switch`.

```
623         /* El disco es correcto. Falta dar de baja la transmision */
624         if (! terminaTrans(cualDisco, cons) )
625             System.out.println("No se realizo la baja de la transmision");
626         else
627             System.out.println("Transmision terminada");
628         break;
```

#### MSTRA\_UNDISCO :

1. Nuevamente muestras al usuario los discos disponibles usando el método que programaste para ello, que recibe el encabezado del listado.

```
634     case MSTRA_UNDISCO:
635         System.out.println( mstraCatalogo( "Discos_disponibles_"
636                                             + "en_el_catalogo" ) );
```

2. A continuación, usando el método pideNum solicitas al usuario digite la posición del disco deseado, registrando la opción en la variable cualDisco, que ya tienes declarada local al método.

```
637         cualDisco = pideNum(cons, "Elige_el_numero_de_disco,",
638                               0, numDiscos - 1);
```

3. Como estás recibiendo datos arbitrarios, verificas que el disco solicitado esté en rango. En caso que no sea así, sales del switch y regresas a verificar la condición del while.

```
639         if (cualDisco == -1 ) {
640             System.out.println("El_disco_elegido_no_existe");
641             break;
642         }
```

4. Si sigues dentro del bloque del switch, vas a copiar la referencia al disco seleccionado en una variable de tipo Disco.

```
643         elDisco = catalogo[cualDisco];
```

Esta variable la declaras al inicio del método, a continuación de la declaración de cualDisco.

```
540     Disco elDisco;           // Para guardar referencias a un disco
```

5. Verificas que el disco exista para no usar una referencia nula. Si no existe, das el mensaje adecuado y sales del switch.

```
644         if (elDisco == null) {
645             System.out.println("Ese_disco_no_existe");
646             break;
647         }
```

Puedes notar que la ejecución va a pasar más tiempo verificando datos que trabajando propiamente dicho, pero es indispensable que la ejecución no aborte simplemente por un usuario que no proporciona bien los datos.

6. Ahora sí muestras el disco, usando para ello el método correspondiente de la clase del catálogo y sales del switch para continuar con la iteración.

```
648         System.out.println( elDisco.muestraDisco
649                               ("[" + cualDisco + "]_Disco"));
650         break;
```

**MSTRA\_HIST** : En este caso se trata de mostrar al usuario las transmisiones iniciadas y terminadas de un disco dado, el histórico del disco.

1. Le muestras al usuario el catálogo y le pides que elija un disco, verificando como siempre que la elección sea correcta.

```
652     case MSTRA_HIST:
653         System.out.println(mstraCatalogo("Discos disponibles en_"
654                                           + "el catalogo"));
655         cualDisco = pideNum(cons, "Elige el disco del que quieres_"
656                               + "el historico,", 0, numDiscos - 1);
657         if (cualDisco == -1) {
658             System.out.println("Numero incorrecto de disco");
659             break;
660         }
```

2. Si el disco elegido existe, verificas que tenga registros históricos (que se haya iniciado y terminado una transmisión). Si no tiene histórico, das el mensaje correspondiente y regresas al final de la iteración.

```
663         if (numHist[cualDisco] == 0) {
664             System.out.println("El disco[" + cualDisco
665                               + "] no tiene historia");
666             break;
667         }
```

3. Si sigues en esta opción, usas el método que ya programaste, dentro de un `println` para imprimir la cadena que te entrega y regresas al final de la iteración.

```
668         System.out.println(mstraHist(cualDisco));
669         break;
```

**MSTRA\_HISTRs** : Esta opción muestra todo el histórico que tenga guardado el catálogo. Invocas a la función que programaste para ello e imprimes la cadena que este método te entrega.

```
671     case MSTRA_HISTRs:
672         System.out.println(mstraHistrs ());
673         break;
```

**default** : Esta opción se va a seleccionar en caso de que el usuario haya tecleado una opción no válida, por lo que simplemente le avisas que esta opción no está implementada. Como esta es la última opción, cierras el `switch`.

```
675     default:
676         System.out.println("Opcion no implementada\n");
677 } // switch
```

Una vez terminada la lista de opciones, terminas el `do ... while` pidiendo que la última opción seleccionada no sea -1. A continuación de la iteración termina el método.

```
678     } while (opcion != 0); // do ... while
679 } // conectaCatlgo
```

Con esto terminaste el método `conectaCatlgo` y lo que sigue es probar tu clase, lo que harás en el siguiente video en el método `main` de la clase `Catalogo`.