



<b><i>TOP MUSIC</i></b>
-------------------------

<https://github.com/susomejias>

Fecha entrega: 13-02-2017  
Autores: Jesús Mejías Leiva

1) Creo la canción y añado al top.

```
/**
 * Añade canción al top
 *
 * @param top
 */
private static void anadirCancion(TopMusic top) {
    if (top.anadir(Teclado.pedirEnteroPositivo("Introduce la posición de la canción a introducir.") - 1,
        new Cancion(Teclado.leerCadena("Título de la canción: "),
            Teclado.leerCadena("Introduce el autor de la canción"),
            Teclado.pedirEnteroPositivo("Introduce el año de publicación de la canción"))) == false) {
        System.out.println("Error la canción no se pudo añadir!!");
    } else {
        System.out.println("La canción se añadió, correctamente");
    }
}
```

2) Evitas el subir del top 1 (la primera canción no puede subir más) (LEY DE DEMETER).

```
/**
 * Sube puesto en el top
 *
 * @param indice
 * @return false
 * @return true
 */
public boolean subirPuesto(int indice) {
    //index < 0 || index >= size()
    if (indice <= 0 || indice >= top.size()) {
        return false;
    }

    //top.set(indice - 1, top.get(indice));
    top.add(indice - 1, top.remove(indice));
    return true;
}
```

3) Evitas el bajar la última canción (la última canción no puede bajar más) (LEY DE DEMETER).

```
/**
 * Baja puesto en el top
 *
 * @param indice
 * @return false
 * @return true
 */
public boolean bajarPuesto(int indice) {
    if (indice <= 0 || indice >= top.size()) {
        return false;
    }
    top.add(indice + 1, top.remove(indice));
    return true;
}
```

4)Obtienes el top de la canción. (LEY DE DEMETER).

```
/**
 * Devuelve la canción mas escuchada
 *
 * @param indice
 * @return top.get(0)
 */
public Cancion CancionMasEscuchada() {
    return top.get(0);
}
```

5)Muestras los 10 mejores. (LEY DE DEMETER).

```
@Override
public String toString() {
    String cadena = "";
    int contador = 1;
    for (Cancion cancion : top) {
        cadena += ">>TOP" + contador++ + ">> " + cancion + "\n";
    }
    return cadena;
}
```

6)Borras la canción (LEY DE DEMETER).

```
/**
 * Elimina elemento del top
 *
 * @param elemento
 * @return false
 * @return true
 */
public boolean eliminar(int indice) {
    if (top.isEmpty() == true || indice < 0 || indice > top.size()) {
        return false;
    }

    top.remove(indice - 1);
    return true;
}
```

7)Te sales del programa.

```
case 0:
    System.exit(0);
```

8)Le indicas al usuario que la lista está vacía, por lo que no puedes ni subir, ni bajar, ni eliminar...(PRINCIPIO DRY).

```
boolean isEmpty() {
    if (top.size() <= 0) {
        return true;
    }
    return false;
}
```

9)Al añadir, no le pides al usuario los datos de la canción si la posición es incorrecta (USABILIDAD).

```
/**
 * Añade elemento al top
 * @param elemento
 * return true
 * return false
 */
boolean anadir(int indice, Cancion elemento) {
    if (indice < 0 || indice > top.size())
        return false;

    top.add(indice - 1, elemento);
    return true;
}
```

10)Al añadir, le pides al usuario la posición de la canción. Recuerda que el usuario dará la posición 1, mientras que internamente será un add(0, cancion)... (USABILIDAD).

```
/**
 * Añade elemento al top
 * @param elemento
 * return true
 * return false
 */
boolean anadir(int indice, Cancion elemento) {
    if (indice < 0 || indice > top.size())
        return false;

    top.add(indice - 1, elemento);
    return true;
}
```