

**Pregunta 5** (6 puntos)

Se tiene la clase Main donde se han definido 6 pruebas para la clase ListaDeReproduccion como se ve en la Columna "A". Cada prueba devuelve un booleano que indica si ha sido superada. Se usa además la clase Cancion, sólo con funcionalidades limitadas a lo que se necesita en las pruebas de **ListaDeReproduccion**.

Usando las 2 primeras pruebas, se ha programado la clase ListaDeReproduccion como se ve en la Columna "B". Se pide:

- a) **(1.5 puntos)** Programar el método "**agregarCancion** " y ajustar el método "**cuales**" a fin de que se pase la prueba 3 (test3) sin romper las pruebas anteriores
- b) **(1.5 puntos)** Ajustar los métodos "**agregarCancion**" y "**cuales**" a fin de que se pase la prueba 4 (test4) sin romper las pruebas anteriores.
- c) **(1.5 puntos)** Ajustar los métodos "**agregarCancion**" y "**cuantoDura**" a fin de que se pase la prueba 5 (test5) sin romper las pruebas anteriores.
- d) **(1.5 puntos)** Ajustar el método "**agregarCancion**" y programar el método "**cuantas**" a fin de que se pase la prueba 6 (test6) sin romper las pruebas anteriores.

Nota: Puede usar los números de línea para indicar dónde se insertaría el nuevo código de la clase Menu y/o si se debe cambiar o retirar alguna línea del código original de la clase Menu. La sintaxis tiene que hacer sentido (no es estricta).

Columna A	Columna B
<pre> public class Main{     public static void main(String args[])     {         Main pruebas = new Main();         pruebas.correrPruebas();     }      public void correrPruebas()     {         System.out.println("Prueba 1:"+test1());         System.out.println("Prueba 2:"+test2());         System.out.println("Prueba 3:"+test3());         System.out.println("Prueba 4:"+test4());         System.out.println("Prueba 5:"+test5());     }      public boolean test1()     {         ListaDeReproduccion milista=new ListaDeReproduccion();         return (milista.cuantoDura()==0);     }      public boolean test2()     {         ListaDeReproduccion milista=new ListaDeReproduccion();         String salida=milista.cuales();         return salida.equals("No hay canciones");     }      public boolean test3()     {         ListaDeReproduccion milista=new ListaDeReproduccion();         Cancion cancion1=new Cancion("Color de Esperanza",260);         milista.agregarCancion(cancion1);         String salida=milista.cuales();         return salida.equals("1. Color de Esperanza");     }      public boolean test4()     {         ListaDeReproduccion milista=new ListaDeReproduccion();         Cancion cancion1=new Cancion("Color de Esperanza",260);         milista.agregarCancion(cancion1);         Cancion cancion2=new Cancion("Pies Descalzos",206);         milista.agregarCancion(cancion2);         String salida=milista.cuales();         return salida.equals("1. Color de Esperanza, 2. Pies Descalzos");     }      public boolean test5()     {         ListaDeReproduccion milista=new ListaDeReproduccion();         Cancion cancion1=new Cancion("Color de Esperanza",260);         milista.agregarCancion(cancion1);         Cancion cancion2=new Cancion("Pies Descalzos",206);         milista.agregarCancion(cancion2);         return (milista.cuantoDura()==466);     }      public boolean test6()     {         ListaDeReproduccion milista=new ListaDeReproduccion();         Cancion cancion1=new Cancion("Color de Esperanza",260);         milista.agregarCancion(cancion1);         Cancion cancion2=new Cancion("Pies Descalzos",206);         milista.agregarCancion(cancion2);         return (milista.cuantas()==2);     } }  class Cancion{     String titulo;     float duracion;      Cancion(String titulo, float duracion)     {         this.titulo=titulo;         this.duracion=duracion;     } } </pre>	<pre> 1 class ListaDeReproduccion 2 { 3     String[] canciones = new String[10]; 4     float duracion; 5 6     public float cuantoDura() { 7         return 0; 8     } 9 10    public String cuales() { 11        return "No hay canciones"; 12    } 13 } </pre>