RETO 2 – PROGRAMACIÓN BÁSICA

CONTEXTO

La casa desarrolladora UdeASoft desea desarrollar un nuevo reproductor de música.

Usted ha sido contratado como Desarrollador Experto en Java, porque ha logrado demostrar habilidades de desarrollo en este lenguaje de programación y se le ha concedido implementar la clase correspondiente al reproductor de música.

Esta clase contará con las siguientes funcionalidades:

- Reproducir siguiente canción: Cambia la canción actual que se está reproduciendo por la siguiente en la lista; en caso tal de que la canción actual que se está reproduciendo sea la última, se vuelve a reproducir la primera canción.
- Reproducir canción anterior: Cambia la canción actual que se está reproduciendo por la anterior en la lista; en caso tal de que la canción actual que se está reproduciendo sea la primera, el reproductor procede a reproducir la última canción.
- Cambiar el estado de reproducción de una canción: Reproducir o pausar una canción; si la canción se está reproduciendo, la canción se pausará cuando el usuario lo indique, y si por el contrario la canción está pausada, esta se seguirá reproduciendo.

  Nota: Si el reproductor está pausado, esto **NO** impedirá cambiar de canción a la siguiente o a la anterior, ni tampoco impedirá agregar canciones a la categoría de favoritas.

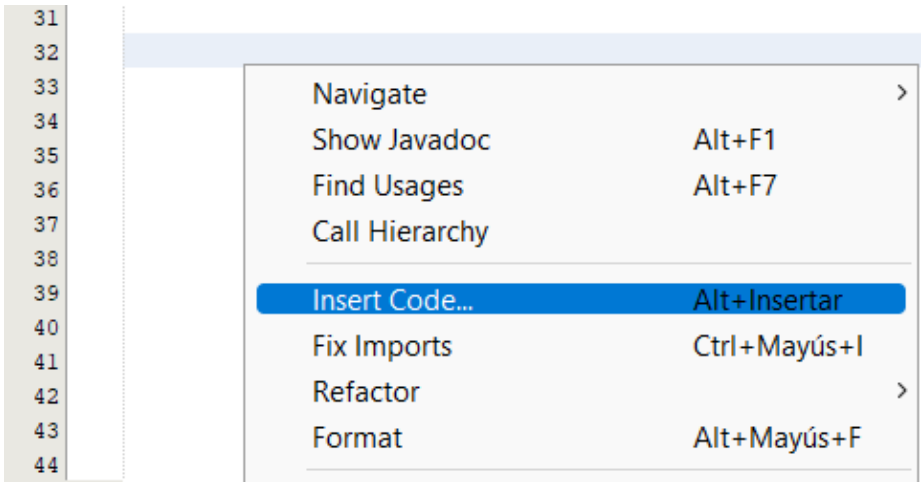
  Si el reproductor está pausado y se pasa a la siguiente canción o se devuelve a la canción anterior, el reproductor mantiene el estado de pausado.

- Agregar canciones a la categoría *Canciones favoritas*: Si la canción que se está reproduciendo no está clasificada como favorita, se agrega a la lista de favoritas, si ya lo está, el reproductor ignora la acción.

Para facilitar la implementación de la clase `ReproductorMusica`, el equipo de Ingeniería de software le hace entrega del diagrama de clases (Figura 1).

Recuerde que los métodos relacionados al constructor, getters y setters son obviados en el diagrama de clases, pero deberán ser incluidos en el código; estos métodos deben ser creados con el estándar camel case, por ejemplo, si el atributo se llama `cancionReproduciendo`, sus métodos correspondientes a get y set serían `getCancionReproduciendo` y `setCancionReproduciendo`, para el caso de los atributos de tipo `boolean`, el `get` se cambia por un `is`, por ejemplo, si el atributo se llama `pausado` y es de tipo `boolean`, su getter será `isPausado`.

Nota: Recuerde que desde NetBeans puede generar automáticamente los getter y setters con la opción `Insert Code...`



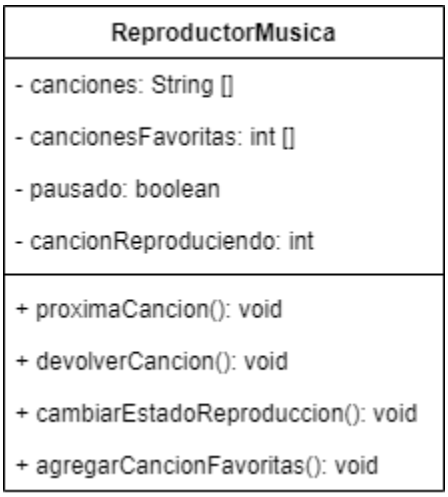


Figura 1

Además del diagrama, el equipo de Ingeniería entrega esta documentación para comprender mejor los elementos del diagrama:

Clase ReproductorMusica

Atributos

NOMBRE	TIPO DATO	CONCEPTO	INICIALIZACIÓN
`canciones`	`String[]`	Contiene los nombres de todas las canciones que el usuario ha adquirido	En el método constructor
`cancionesFavoritas`	`int[]`	Contiene los índices de todas las canciones que el usuario ha clasificado en la categoría de *Canciones favoritas*	En el método constructor se inicializa con longitud igual a `canciones` y con -1 en todas las posiciones
`pausado`	`boolean`	Guarda `true` si el reproductor está pausado, y `false` si no lo está	Debe estar inicializada en `true`
`cancionReproduciendo`	`int`	Guarda la posición del array `canciones` en la que se encuentra la canción que se está reproduciendo	Debe estar inicializada en 0

Métodos (Ningún método recibe parámetros, y todos retornan `void`)

NOMBRE	CONCEPTO
`proximaCancion`	Actualiza el valor de `cancionReproduciendo` por `(cancionReproduciendo + 1)%canciones.length` (La operación módulo garantiza que cuando se esté reproduciendo la última canción y se use este método, se reproduzca la primera canción del array)

`devolverCancion`	Actualiza el valor de `cancionReproduciendo` por `(cancionReproduciendo + canciones.length - 1)%canciones.length` (La operación módulo garantiza que cuando se esté reproduciendo la primera canción y se use este método, se reproduzca la última canción del array)
`cambiarEstadoReproduccion`	Actualiza el valor de `pausado` por su valor booleano contrario (Si está en `false`, lo pasa a `true`; si está en `true`, lo pasa a `false`)
`agregarCancionFavoritas` **(Leer Nota 1)**	Agrega la posición de la canción que está en reproducción a `cancionesFavoritas` si aún no está en la lista, en caso contrario, no el reproductor no hará nada más. (La canción que se agregue, se agrega en la primera posición vacía que encuentre)

NOTA 1

El método **agregarCancionFavoritas YA** se le entrega **implementado** en la plantilla del VPL, no se debe preocupar por desarrollarlo.

TAREAS

- En el archivo preconstruido en la plataforma Moodle, implementar la clase especificada en el diagrama de clases, teniendo en cuenta la documentación dada por el equipo de Ingeniería de software.
- Los nombres de los métodos y atributos **DEBEN** ser nombrados tal y como aparecen en el diagrama de clases.
- Usted **NO** debe solicitar datos por teclado, ni programar un método `main`, tampoco debe especificar el paquete al que pertenece la clase, usted está solamente encargado de la construcción de la clase.

EJEMPLO

El calificador hará veces de usuario y será quien evalúe el funcionamiento del reproductor que usted desarrolló:

1. El calificador añade diecisiete (17) canciones a su reproductor:

```
String [] cancionesEscogidas = new String [] {
    "Mil horas",
    "Por ese hombre",
    "A esa",
    "Algo de mi",
    "Si me dejas ahora",
    "¿Quieres ser mi amante?",
    "Quel sorriso in volto",
    "Per una notte insieme",
    "Como un pintor",
    "Sencillamente",
    "Un segundo",
    "Enciéndeme",
    "Cuando me enamoro",
    "Tu eres mi tesoro",
    "Piccola anima",
    "Solo a ti pertenezco",
    "Todos por todos"
};

ReproductorMusica ventana1 = new ReproductorMusica(cancionesEscogidas);
```

Note que usted **NO** añade las canciones, lo hace quien use su clase (En este caso el calificador).

Por lo tanto, se espera que los atributos del objeto `ventana1` tengan los siguientes atributos:

```
Canciones: ["Mil horas", "Por ese hombre", "A esa", "Algo de mi",
"Si me dejas ahora", "¿Quieres ser mi amante?", "Quel sorriso in volto", "Per una notte insieme",
"Como un pintor", "Sencillamente", "Un segundo", "Enciéndeme",
"Cuando me enamoro", "Tu eres mi tesoro", "Piccola anima", "Solo a ti pertenezco",
Todos por todos]

Canciones favoritas: [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]

Pausado: true
Canción en reproducción: 0
```

2. El calificador usa el método `devolverCancion` cuatro (4) veces:

```
ventana1.devolverCancion();
ventana1.devolverCancion();
ventana1.devolverCancion();
ventana1.devolverCancion();
```

Por lo tanto, se espera que los atributos del objeto `ventana1` se actualicen de manera que queden de la siguiente manera:

```
Canciones: ["Mil horas", "Por ese hombre", "A esa", "Algo de mi",
"Si me dejas ahora", "¿Quieres ser mi amante?", "Quel sorriso in volto", "Per una notte insieme",
"Como un pintor", "Sencillamente", "Un segundo", "Enciéndeme",
"Cuando me enamoro", "Tu eres mi tesoro", "Piccola anima", "Solo a ti pertenezco",
Todos por todos]

Canciones favoritas: [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]

Pausado: true
Canción en reproducción: 13
```

Como se esperaba, la canción en reproducción se devolvió cuatro (4) veces, cuando el calificador devolvió la canción la primera vez, `cancionReproduciendo` quedó en posición 16; en la segunda vez que la devuelve, quedó en la posición 15; en la tercera vez que la devuelve, quedó en la posición 14; y en la última vez que la devuelve, quedó en la posición 13.

3. El calificador realiza las siguientes operaciones en ese orden:

```
ventana1.agregarCancionFavoritas();
ventana1.proximaCancion();
ventana1.agregarCancionFavoritas();
ventana1.devolverCancion();
ventana1.agregarCancionFavoritas();
```

Por lo tanto, se espera que los atributos del objeto `ventana1` se actualicen de manera que queden de la siguiente manera:

```
Canciones: ["Mil horas", "Por ese hombre", "A esa", "Algo de mi",
"Si me dejas ahora", "¿Quieres ser mi amante?", "Quel sorriso in volto", "Per una notte insieme",
"Como un pintor", "Sencillamente", "Un segundo", "Enciéndeme",
"Cuando me enamoro", "Tu eres mi tesoro", "Piccola anima", "Solo a ti pertenezco",
Todos por todos]

Canciones favoritas: [13, 14, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]

Pausado: true
Canción en reproducción: 13
```

En el inciso anterior, la canción en reproducción era la ubicada en la posición 13, en este inciso, el calificador la agrega a las canciones favoritas; el calificador pasa a la siguiente canción (Canción con índice 14) y también la agrega a canciones favoritas; luego se devuelve a la canción anterior (canción en la posición 13) y la intenta agregar nuevamente a las canciones favoritas, como **YA** está, **NO** se vuelve a

agregar, usted **debe** mantener el orden en el que se van agregando al array de canciones favoritas.

4. El calificador realiza las siguientes operaciones en ese orden:

```
ventana1.cambiarEstadoReproduccion();
ventana1.cambiarEstadoReproduccion();
ventana1.cambiarEstadoReproduccion();
ventana1.cambiarEstadoReproduccion();
ventana1.cambiarEstadoReproduccion();
```

En estas cinco líneas de código, el calificador cambia 5 veces el estado de la reproducción, es decir, inicialmente está en `pausado=true`, la modificación se haría en este orden: `false`, `true`, `false`, `true`, `false` (5 cambios). Por lo tanto, se espera que el atributo `pausado` del objeto `ventana1` se actualice quedando de la siguiente manera:

```
Canciones: ["Mil horas", "Por ese hombre", "A esa", "Algo de mi",
"Si me dejas ahora", "¿Quieres ser mi amante?", "Quel sorriso in volto", "Per una notte insieme",
"Como un pintor", "Sencillamente", "Un segundo", "Enciéndeme",
"Cuando me enamoro", "Tu eres mi tesoro", "Piccola anima", "Solo a ti pertenezco",
Todos por todos]

Canciones favoritas: [13, 14, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]

Pausado: false
Canción en reproducción: 13
```

Si desea realizar el ejemplo en Netbeans, copia y pega el siguiente código dentro de tu método `main`:

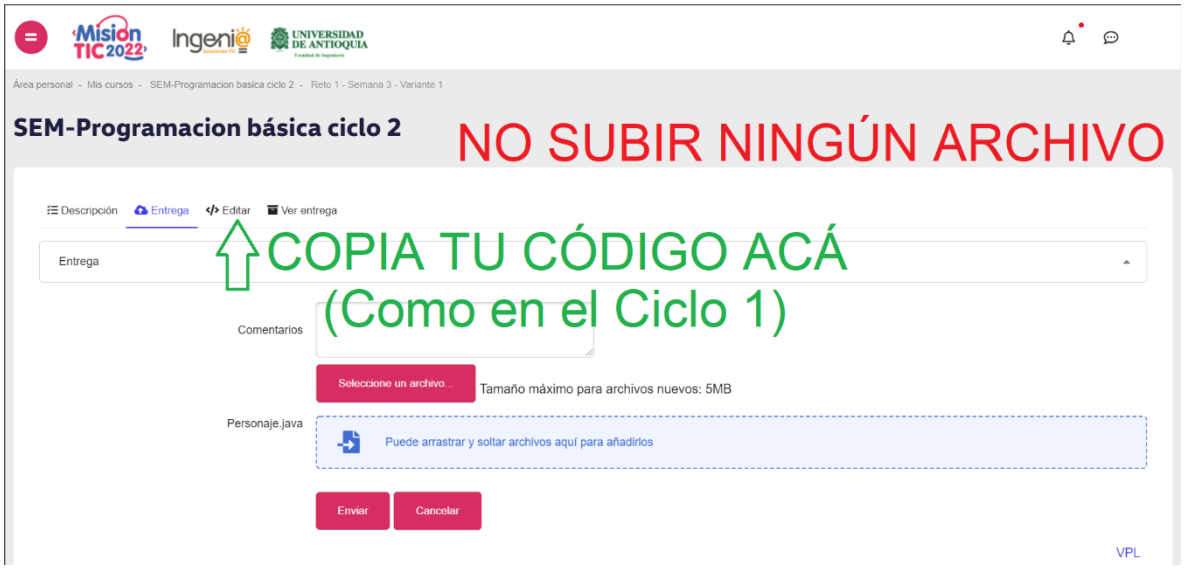
```
//Screenshot 1

String [] cancionesEscogidas = new String [] {

    "Mil horas",

    "Por ese hombre",

    "A esa",

    "Algo de mi",

    "Si me dejas ahora",

    "¿Quieres ser mi amante?",

    "Quel sorriso in volto",

    "Per una notte insieme",

    "Como un pintor",
```

```
    "Sencillamente",
    "Un segundo",
    "Enciéndeme",
    "Cuando me enamoro",
    "Tu eres mi tesoro",
    "Piccola anima",
    "Solo a ti pertenezco",
    "Todos por todos"
};
ReproductorMusica ventana1 =new ReproductorMusica(cancionesEscogidas);
//Screenshot 2
ventana1.devolverCancion();
ventana1.devolverCancion();
ventana1.devolverCancion();
ventana1.devolverCancion();

//Screenshot 3
ventana1.agregarCancionFavoritas();
ventana1.proximaCancion();
ventana1.agregarCancionFavoritas();
ventana1.devolverCancion();
ventana1.agregarCancionFavoritas();

//Screenshot 4
ventana1.cambiarEstadoReproduccion();
ventana1.cambiarEstadoReproduccion();
ventana1.cambiarEstadoReproduccion();
ventana1.cambiarEstadoReproduccion();
ventana1.cambiarEstadoReproduccion();
```

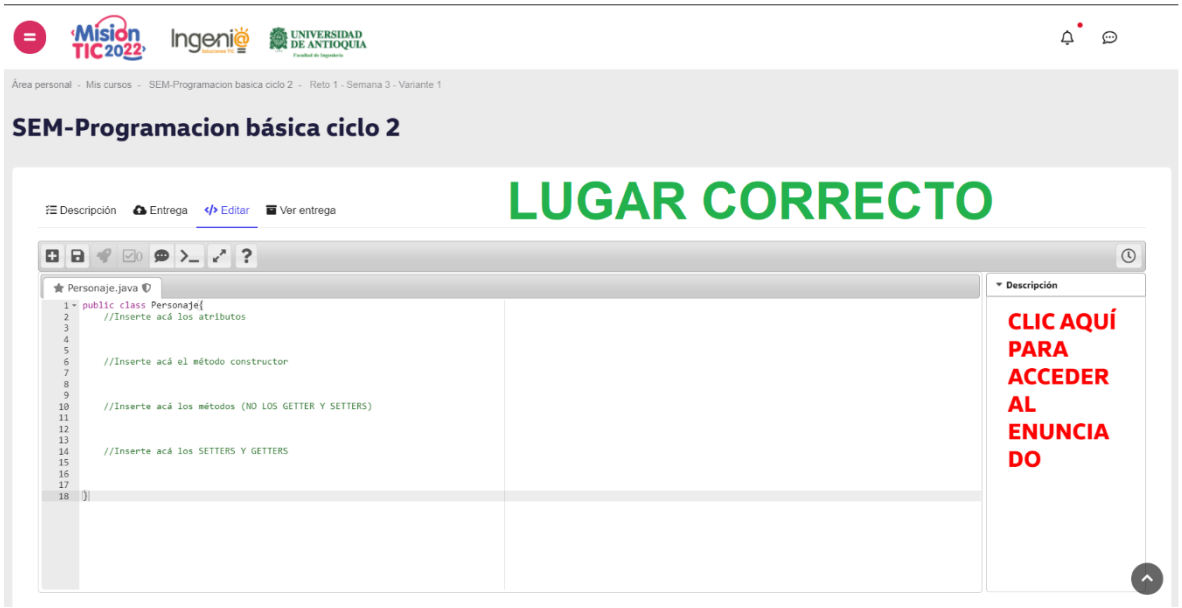
NOTA ACLARATORIA

Usted podrá desarrollar la clase requerida en un IDE como NetBeans, y al final copiar y pegar el código en la herramienta VPL, pero **NO** deberá subir archivos, es decir:

Modo incorrecto:



Modo correcto:



¡MUCHOS ÉXITOS EN EL DESARROLLO DEL RETO 2 TRIPULANTE!

ACLARACIÓN DE PLAGIO

El objetivo es que los tripulantes cuenten con una oportunidad de aprendizaje relacionada con la programación. La colaboración académica es buena mientras no se lleve a un engaño académico, ya que el engaño académico inflige las buenas conductas del saber y del aprendizaje. El engaño académico hace referencia al plagio o envío de ideas que no son propias.

Colaborar implica compartir ideas, explicar a alguien cómo podría hacer su trabajo (más no hacer el trabajo por el otro) y ayudar al otro si tienes problemas a la hora de ejecutar o encontrar errores en el código.

En aras de evitar el plagio se recomienda colaborar pero no compartir su código o proyecto, no compartir sus soluciones, no usar un código encontrado en internet u otras fuentes que las propias. (Mason, Gavrilovska, y Joyner, 2019)

Los ejercicios enviados a verificación deben cumplir con la política antiplagio. Es decir, cualquier envío que sea una copia textual de otro trabajo puede ser suspendido o no aprobado por parte del equipo evaluador. El acto de copiar material de otro estudiante es un comportamiento inaceptable para el desarrollo de las competencias individuales y su progreso en este curso.

Referencia.

Mason, T., Gavrilovska, A., y Joyner, D.A. (2019). *Collaboration vesus cheating. 50th ACM Technical Symposium on Computer Science Education SIGCSE 2019*, Mineapolis, MN. DOI: 10.1145/3287324.3287443