

Se quiere emular el funcionamiento de un reproductor de música. Para ello se va a disponer de dos clases. Una que va a representar la canción y otra que va a representar al reproductor.

The screenshot shows two code editor panes. The top pane contains the definition of the `Cancion` class, which has attributes `titulo` and `duracion`, and methods `reproducir_cancion`, `__eq__`, `__str__`, `no_reproducciones`, and `__no_reproducciones`. The bottom pane contains the definition of the `ReproductorMusica` class, which has attribute `canciones` and methods `__init__`, `__str__`, `insertar_canciones`, `reproducir_cancion_seleccionada`, `avanzar_puntero`, `hay_siguiente`, `__buscar_cancion`, `__buscar_cancion1`, `__buscar_cancion2`, `seleccionar_cancion`, `borrar_cancion`, and `seleccionar_canciones_mas_escuchadas`.

```
class Cancion:
    def __init__(self, titulo: str, duracion: float):
        self.titulo = titulo
        self.duracion = duracion

    def reproducir_cancion(self):
        pass

    def __eq__(self, other):
        return self.titulo == other.titulo and self.duracion == other.duracion

    def __str__(self):
        return f'Cancion: {self.titulo} - {self.duracion}'

    def no_reproducciones(self):
        pass

    def __no_reproducciones(self, no_reproducciones: int):
        pass

class ReproductorMusica:
    def __init__(self):
        self.canciones = []

    def __str__(self):
        pass

    def insertar_canciones(self, canciones: List[Cancion]):
        self.canciones.extend(canciones)

    def reproducir_cancion_seleccionada(self):
        pass

    def avanzar_puntero(self):
        pass

    def hay_siguiente(self):
        pass

    def __buscar_cancion(self, titulo: str, duracion: int):
        pass

    def __buscar_cancion1(self, titulo: str, duracion: int):
        pass

    def __buscar_cancion2(self, titulo: str, duracion: int):
        pass

    def seleccionar_cancion(self, titulo: str, duracion: int):
        pass

    def borrar_cancion(self, titulo: str, duracion: int):
        pass

    def seleccionar_canciones_mas_escuchadas(self, veces: int = 2):
        pass
```

Figura 1 Clases del reproductor

Partiendo del código de apoyo se deben implementar los siguientes métodos de `ReproductorMusica`:

- `reproducir_cancion_seleccionada`: reproduce la canción a la que apunta el puntero
- `avanzar_puntero`: avanza el puntero a la siguiente canción de la lista, si el puntero no está al final de la lista.
- `hay_siguiente`: Indica si el puntero está en la última canción de la lista
- `__buscar_cancion`: Si hay una canción con el título y la duración dadas, devuelve su posición en otro caso devuelve -1. En la figura aparecen varios buscar. Son diversas implementaciones del mismo método. El alumno sólo debe proporcionar una implementación.

- **seleccionar\_cancion:** Si la canción existe, pone el puntero en la primera ocurrencia de la canción con el título y la duración dados en otro caso deja el puntero igual
- **borrar\_cancion:** Si la canción dada está en la biblioteca, borra la primera ocurrencia de la canción y deja el puntero apuntando a la primera canción, en otro caso no hace nada
- **seleccionar\_canciones\_mas\_escuchadas:** Proporciona la lista con las que han sido escuchadas más de un valor dado