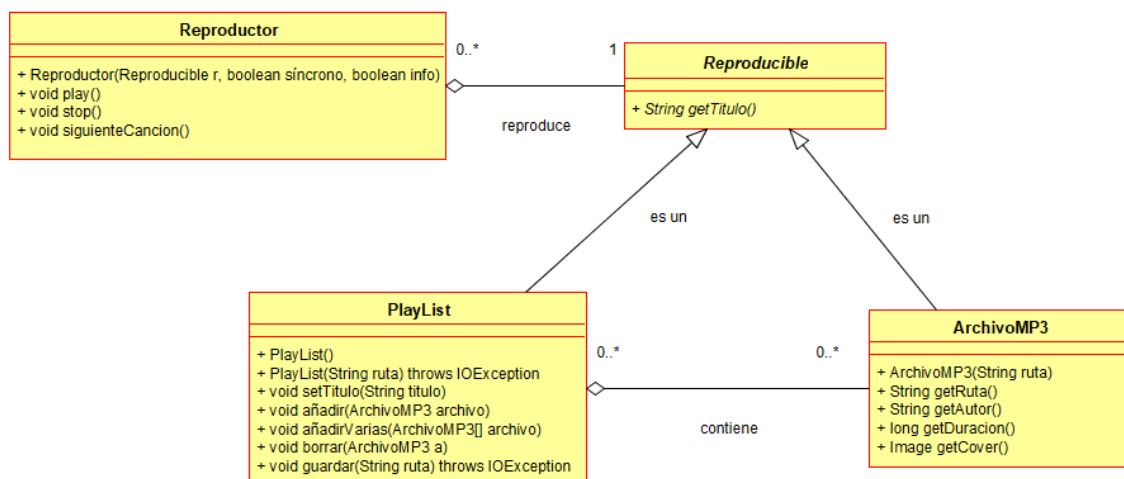


La librería ReproductorMúsica

- **Librería:** *ReproductorMusica*
- **Archivo:** *ReproductorMusica.jar*
- **Desarrollador:** *Juan Diego*
- **Paquete:** *bpc.daw.reproductor*
- **Créditos:** Esta librería usa a su vez las librerías JLayer y mp3agic

Esta librería proporciona una clase Reproductor que es capaz de reproducir archivos en formato MP3. También se permite la creación y utilización de listas de reproducción que pueden ser guardadas en el disco.

El diagrama de clases, que serán explicadas en los sucesivos apartados de este documento es:



1) Reproducible

Esta clase representa un objeto genérico que puede ser reproducido por el Reproductor.

Reproducible
+ String getTitulo()

- **getTitulo:** Devuelve el título asociado al contenido reproducible.

2) ArchivoMP3

Esta clase representa un archivo MP3 que está en alguna ruta del ordenador.

ArchivoMP3
+ ArchivoMP3(String r) + String getRuta() + String getTitulo() + String getAutor() + long getDuracion() + Image getCover()

- **constructor:** Crea un objeto a partir de la ruta que se le pasa como parámetro. Se recomienda usar la barra / para separar las carpetas.
- **getRuta:** Devuelve la ruta del archivo
- **getTitulo:** Devuelve el título de la canción, tal y como aparece en el ID3v2 tag del mp3. Si el archivo no dispone de ese dato, se devuelve el nombre del archivo.
- **getAutor:** Devuelve el autor de la canción, tal y como aparece en el ID3v2 tag del mp3. Si el archivo no dispone de ese dato, devuelve null.
- **getDuración:** Devuelve la duración de la canción. Si el archivo no dispone de ese dato, devuelve -1.
- **getCover:** Devuelve un objeto Image con la imagen de la portada que tienen incluidos algunos archivos mp3. Si el archivo no dispone de esa imagen, devuelve null.

3) Playlist

Esta clase representa una lista de reproducción formada por archivos mp3.

ArchivoMP3
+ Playlist() + Playlist(String ruta) throws IOException + void setTitulo(String titulo) + void getTitulo() + void añadir(ArchivoMP3 archivo) + void añadirVarias(ArchivoMP3... archivos) + void borrar(ArchivoMP3 archivo) + void guardar(String ruta) throws IOException

- **primer constructor:** Crea una lista de reproducción vacía, sin título y sin canciones
- **segundo constructor:** Crea una lista de reproducción vacía con los datos del archivo cuya ruta se pasa como parámetro. Si no puede cargar el archivo, o tiene un formato incorrecto, lanzará una IOException
- **setTitulo:** Cambia el título de la lista de reproducción
- **getTitulo:** Devuelve el título de la lista de reproducción
- **añadir:** Añade un objeto ArchivoMP3 a la lista de reproducción
- **añadirVarias:** Permite añadir varios ArchivoMP3 a la lista de reproducción
- **borrar:** Borra de la lista de reproducción el ArchivoMP3 que se pasa como parámetro
- Si el archivo no dispone de ese dato, devuelve null.
- **guardar:** Guarda la lista de reproducción en un archivo cuya ruta se pasa como parámetro. Si el archivo existe, se sobrescribirá. La extensión del archivo puede ser cualquiera, pero se recomienda que sea .txt. Si se produce un error al guardar (falta de espacio en disco, permisos de escritura, etc) se lanzará una IOException.

4) Reproductor

Esta clase representa un reproductor que puede reproducir cualquier objeto Reproducible

Reproductor
+ Reproductor(Reproducible r, boolean síncrono, boolean info) + void play() + void stop() + void siguienteCancion()

- **constructor:** Crea un reproductor, con los siguientes parámetros:
 - r: Un objeto Reproducible, que será el contenido a reproducir
 - síncrono: Si se pasa true, el reproductor esperará a que finalice la reproducción. Si se pasa false, la reproducción se hará en segundo plano y el programa continuará su ejecución.
 - info: Si se pone true, se irá mostrando en la pantalla el nombre de las canciones que se van reproduciendo.
- **play:** Pone en marcha el reproductor
- **stop:** Detiene la actividad del reproductor (no se hace pausa, sino que finaliza completamente). Este método solo funciona cuando la reproducción se hace en segundo plano
- **siguienteCancion:** Pasa a la siguiente canción. Este método solo funciona cuando la reproducción se hace en segundo plano