

1 MidiSystem

- **Librería:** *Java Sound*
- **Desarrollador:** *Sun Microsystems*
- **Archivo:** *Incluido en el JDK*
- **Paquete:** *javax.sound.midi*

Esta clase sirve para tener acceso al sistema MIDI del ordenador y por tanto, poder obtener objetos para controlar los secuenciadores, sintetizadores y otros dispositivos MIDI conectados al equipo.

MidiSystem
+ static Sequencer getSequencer() throws MidiUnavailableException + static Sequence getSequence(File f)

- **getSequencer:** Obtiene el secuenciador que utiliza Java por defecto. Si no hay ningún secuenciador en el equipo (lo que sería raro, porque el sistema operativo siempre proporciona uno), o si este está ocupado por otro programa, se lanzará una *MidiUnavailableException*
- **getSequence:** Carga la canción MIDI cuya ruta se pasa como parámetro

2 Sequencer

- **Librería:** *Java Sound*
- **Archivo:** *Incluido en el JDK*
- **Desarrollador:** *Sun Microsystems*
- **Paquete:** *javax.sound.midi*

Esta clase representa un secuenciador MIDI. Un secuenciador es un dispositivo que reproduce una canción en formato MIDI y le envía los correspondientes comandos musicales a los instrumentos y dispositivos MIDI a los que se encuentra conectado

Sequencer
+ void open() + void close() + void start() + Transmitter getTransmitter() + void setSequence(Sequence s) throws InvalidMidiDataException

- **open:** Abre el secuenciador, de forma que pueda ser utilizado.
- **close:** Cierra el secuenciador, de forma que libera los recursos del sistema que esté utilizando.
- **getTransmitter:** Devuelve un Transmitter con el que conectar la salida de este secuenciador con la entrada de otro dispositivo Midi
- **setSequence:** Pone en el secuenciador la canción MIDI pasada como parámetro. Si el archivo no tiene formato MIDI se lanzará la excepción.
- **start:** Hace que el secuenciador comience a reproducir la canción que previamente haya sido cargada en él con el método setSequence

3 Sequence

- **Librería:** *Java Sound*
- **Archivo:** *Incluido en el JDK*
- **Desarrollador:** *Sun Microsystems*
- **Paquete:** *javax.sound.midi*

Esta clase representa una canción en formato MIDI, y por tanto, está formada por comandos musicales que se envían a los instrumentos MIDI en el momento adecuado.

Sequence
+ long getMicrosecondsLength()

- **getMicrosecondsLength:** Devuelve la duración de la canción en microsegundos.

4 Transmitter

- **Librería:** *Java Sound*
- **Desarrollador:** *Sun Microsystems*
- **Archivo:** *Incluido en el JDK*
- **Paquete:** *javax.sound.midi*

Esta clase representa un cable con el que un dispositivo MIDI se puede conectar con otro para transmitirle datos.

Transmitter
+ void setReceiver(Receiver dispositivo) + Receiver getReceiver() + void close()

- **setReceiver:** Conecta el cable con el dispositivo que se pasa como parámetro. Una vez conectado, el dispositivo será capaz de recibir todos los comandos MIDI que pasen por el cable.
- **getReceiver:** Devuelve el dispositivo receptor de datos al que está conectado el cable
- **close:** Libera los recursos asociados por el cable. Debe llamarse cuando vaya a dejar de ser utilizado.

5 Receiver

- **Librería:** *Java Sound*
- **Archivo:** *Incluido en el JDK*
- **Desarrollador:** *Sun Microsystems*
- **Paquete:** *javax.sound.midi*

Esta interfaz la implementan las clases que representan dispositivos que pueden recibir comandos MIDI desde otro dispositivo MIDI como un secuenciador, instrumento, etc.

Receiver
+void send(MidiMessage mensaje, long tiempo) + void close()

- **send:** Este método es llamado cada vez que el objeto recibe un mensaje MIDI enviado por algún dispositivo al que está conectado. También recibe el instante de tiempo en el que lo ha recibido
- **close:** Libera los recursos asociados por el Receiver. Debe llamarse cuando vaya a dejar de ser utilizado.

6 MidiMessage

- **Librería:** *Java Sound*
- **Archivo:** *Incluido en el JDK*
- **Desarrollador:** *Sun Microsystems*
- **Paquete:** *javax.sound.midi*

Esta clase es un mensaje MIDI que representa un comando genérico que se le envía a un instrumento MIDI.

MidiMessage

7 ShortMessage

- **Librería:** *Java Sound*
- **Archivo:** *Incluido en el JDK*
- **Desarrollador:** *Sun Microsystems*
- **Paquete:** *javax.sound.midi*

Esta clase es un tipo concreto de mensaje MIDI, que representa un comando musical que se envía a un instrumento.

ShortMessage extends MidiMessage
+ int getChannel() + int getCommand() + getData1() + getData2()

- **getChannel:** Devuelve el número de canal asociado al mensaje. Este número es un número del 0 al 15 y normalmente representa el número del instrumento musical que emite el mensaje
- **getCommand:** Es un número que indica la acción musical que se quiere realizar. Se compara con constantes como estas:

ShortMessage.NOTE_ON	Una tecla ha sido pulsada
ShortMessage.NOTE_OFF	Una tecla se ha soltado

Hay muchas más constantes que indican otras posibilidades de los instrumentos musicales, pero las básicas son ShortMessage.NOTE_ON y ShortMessage.NOTE_OFF

- **getData1** y **getData2:** Devuelven dos datos cuyo significado depende del número que ha devuelto el método getCommand. Por ejemplo, para los comandos NOTE_ON y NOTE_OFF estos métodos devuelven esto:
 - **getData1:** Devuelve la nota musical que se ve afectada por el comando musical representado en el mensaje
 - **getData2:** Devuelve el volumen con el que se verá afectada la nota musical al ejecutar el comando.