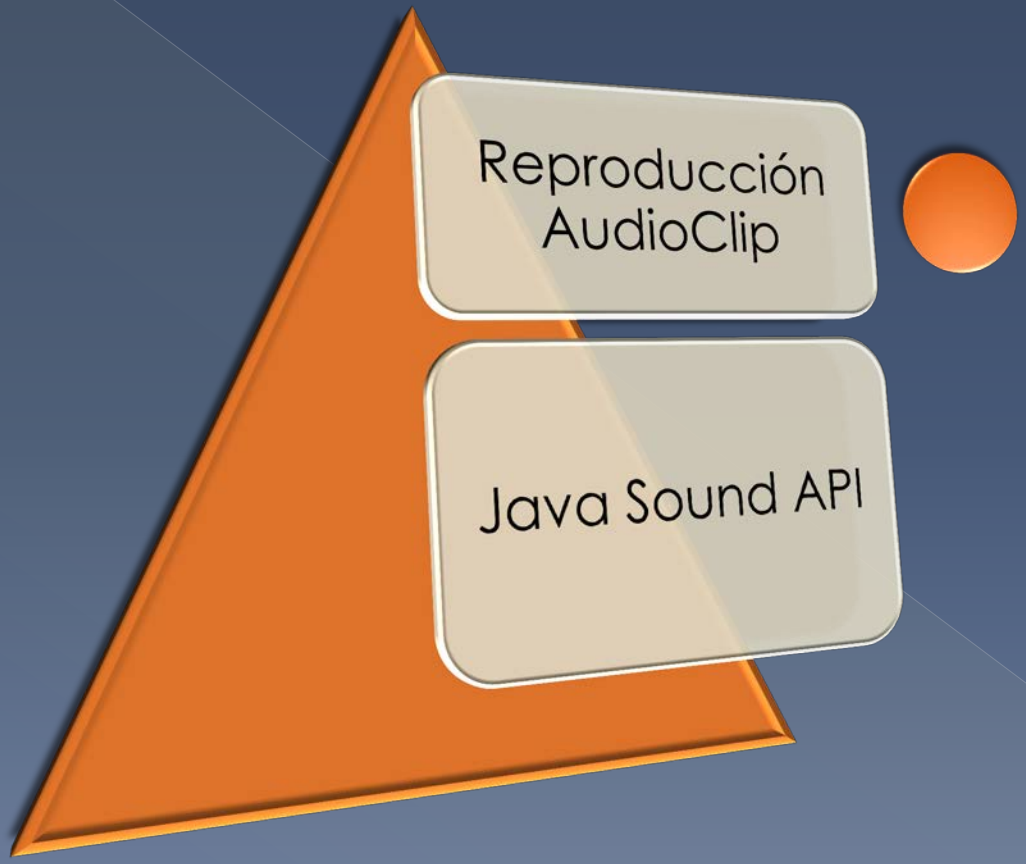
JAVA: Sonido

Alternativas audio en Java

- Reproducción simple
  - Proviene de la primera versión de Java y se basa en el interfaz *AudioClip*
- Java Sound API
  - Permite la captura, mezcla y reproducción de audio, así como control y efecto sobre el sonido
- Avanzada: Java Multimedia Framework
  - *Permite más formatos y mayor control*
  - *Basada en el interfaz Player*

Índice



AudioClip

- Diseñado para la reproducción de audio en applets
- No permite la captura ni la edición
- Reproduce formatos sin compresión basados en PCM (.wave, .au), no reproduce formatos con compresión
- *Métodos: play(), stop(), loop()*

AudioClip

Ejemplo

```
AudioClip audio;

public void play() {
        try{
             URL url = new URL("file:"+" sonido.wav");
             audio = Applet.newAudioClip(url);
             audio.play();
        } catch (Exception e) {
            System.err.println(e);
        }
}

public void stop(){
        audio.stop();
}
```

AudioClip

Ejemplo

```
AudioClip audio;

public void play(File f) {
        try{
             URL url = new URL("file:"+f.getAbsolutePath());
             audio = Applet.newAudioClip(url);
             audio.play();
        } catch (Exception e) {
            System.err.println(e);
        }
}

public void stop(){
        audio.stop();
}
```

AudioClip

Ejemplo

```
AudioClip audio;
URL url = .... ;

public void play() {
        try{
              audio = Applet.newAudioClip(url);
              audio.play();
        } catch (Exception e) {
             System.err.println(e);
        }
}

public void stop(){
        audio.stop();
}
```

Índice

Java Sound API

- API de bajo nivel para controlar la entrada y salida de sonido, tanto de datos de audio y como de datos MIDI.
- Permite manipular recursos del sistema como lectores y escritores de archivos, mezcladores de audio, sintetizadores MIDI, conversores de formato, etc.
- No incluye editores o herramientas gráficas, se centra en el control a bajo nivel

Java Sound API

Paquetes

javax.sound.sampled

javax.sound.sampled.spi

javax.sound.midi

javax.sound.midi.spi

Java Sound API

Paquetes

javax.sound.sampled

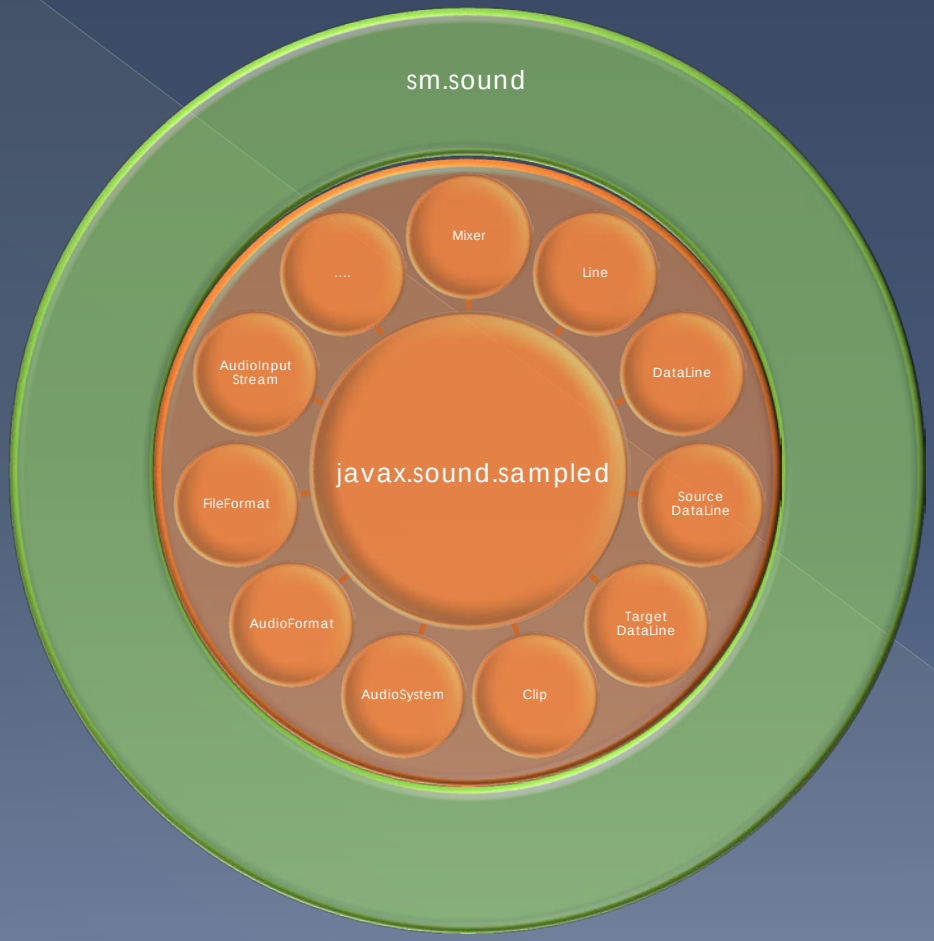
javax.sound.sampled.spi

javax.sound.midi

javax.sound.midi.spi

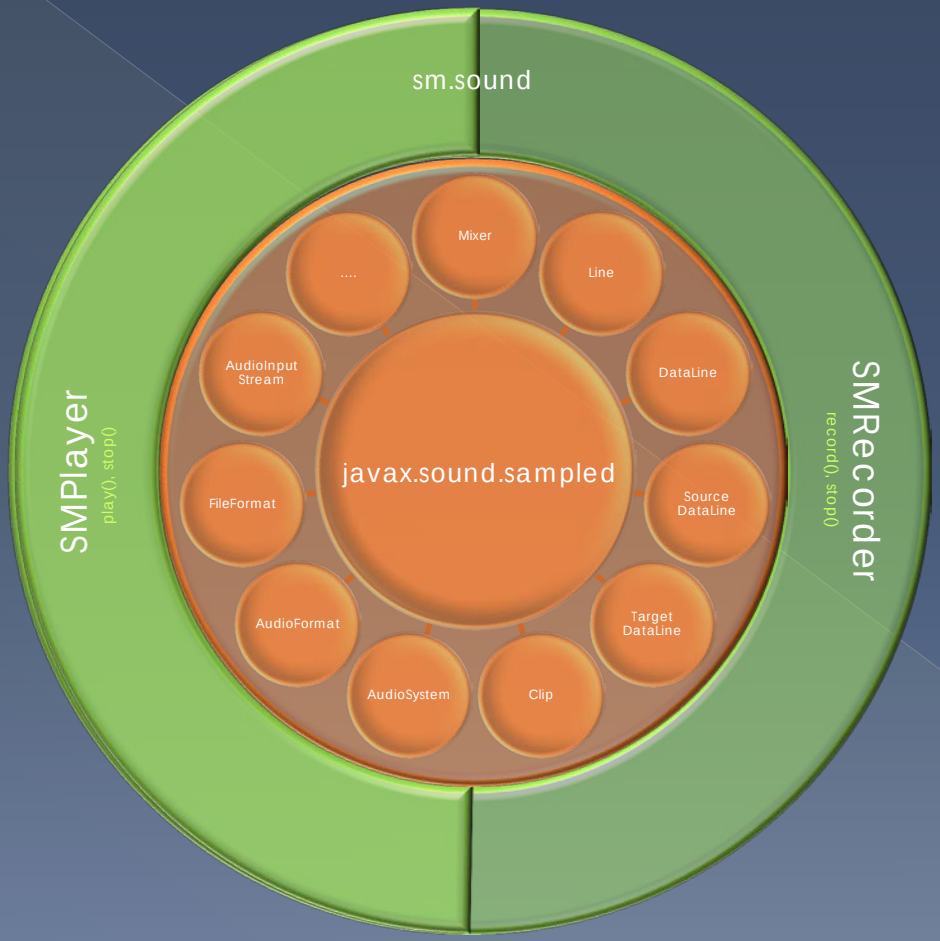
Java Sound API

Paquetes



Java Sound API

Paquetes



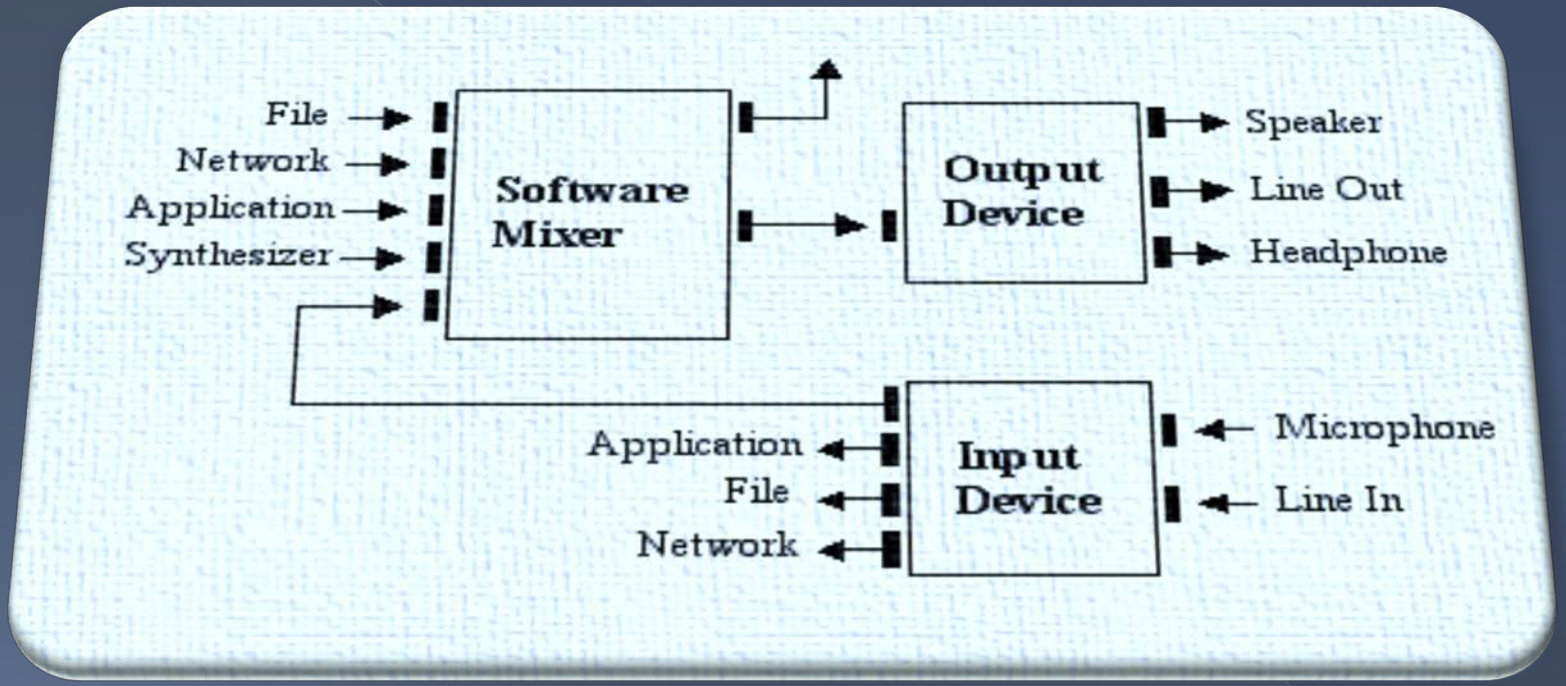
Java Sound API

Paquete javax.sound.sampled

- Permite la reproducción, captura y mezcla de audio, así como control y efecto sobre el sonido, accediendo a los componentes instalados en el sistema
- Trabaja con codificación PCM (sin compresión), μ-law y A-law (compresión sin pérdidas). Por defecto, no considera códec con compresión

Java Sound API

Arquitectura



http://docs.oracle.com/javase/tutorial/sound/index.html

Formatos

Mezcladores

Líneas

Java Sound API

Formatos

- **Formato de datos** indica como interpretar las muestras de audio (codificación, frecuencia de muestreo, resolución, número de canales, etc.).
- **Formato de ficheros**: especifica la estructura de un archivo de sonido (tipo, tamaño, formato de datos, etc.).

AudioFormat

FileFormat

Java Sound API

Mezclador

- Un mezclador gestiona flujos (uno o más) de entrada y salida de audio
- Permite controlar aspectos como el volumen, balance, ganancia, etc.

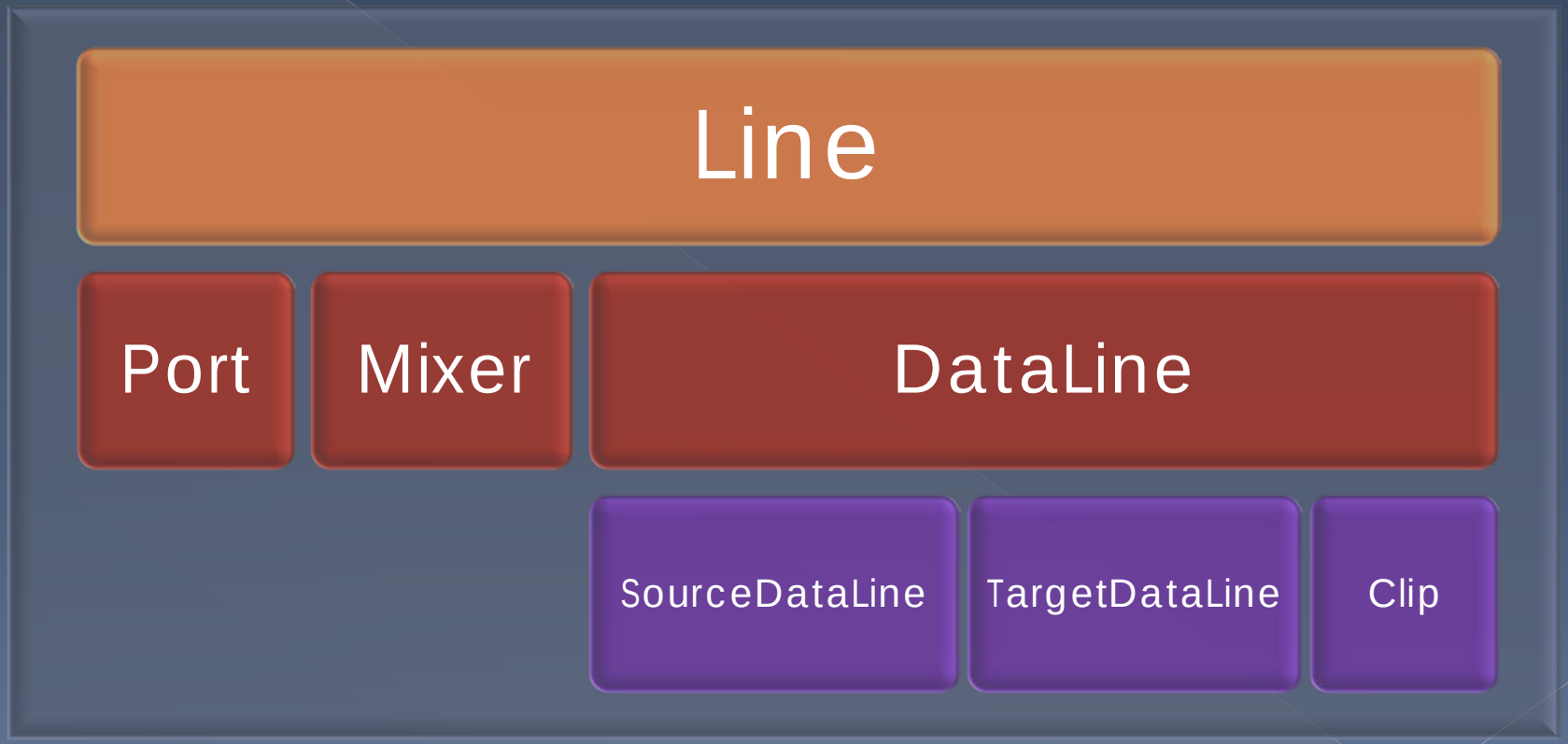
Java Sound API

Línea

- Una línea representan un canal de audio.
- Una línea se puede abrir (reserva recursos del sistema) y cerrar (los libera)
- Tiene asociados controles y genera el evento LineEvent

Java Sound API

Línea



Formatos

Mezcladores

Líneas

Java Sound API

AudioSystem

- La clase *AudioSystem* permite acceder a los recursos del sistema
  - I/O en diferentes formatos de archivo
  - Convertir entre distintos formatos de datos
  - Obtener líneas sin necesidad de mezclador
  - Acceso a mezcladores "por defecto" instalados en el sistema (métodos *getLine*).

Formatos | Mezcladores | Líneas

Java Sound API

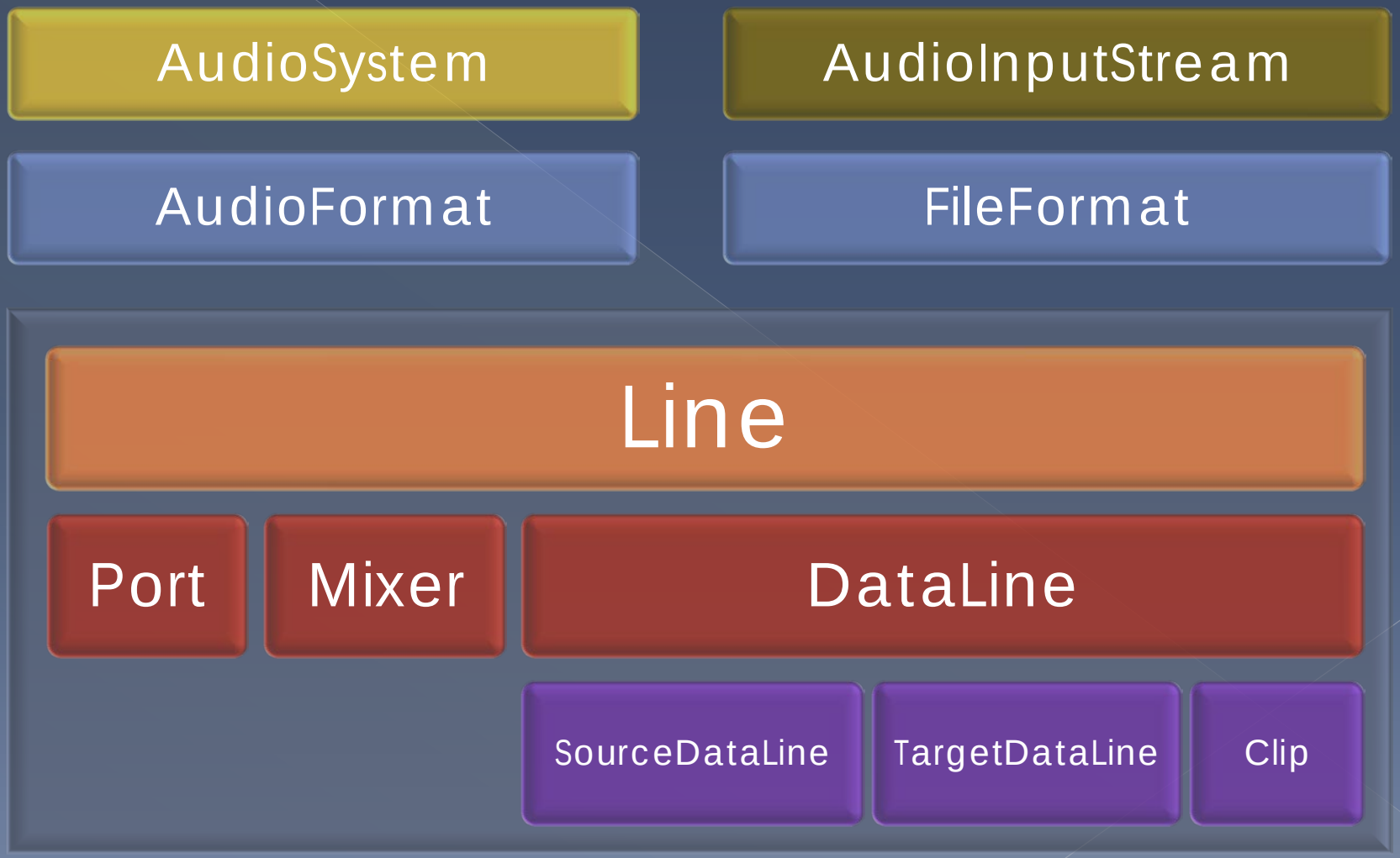
AudioInputStream

- *AudioInputStream* representa un canal de entrada de audio con un determinado formato y longitud
- AudioSystem incluye métodos para manipular AIS (obtener AIS de un fichero, convertir un AIS a otros formatos, etc.)

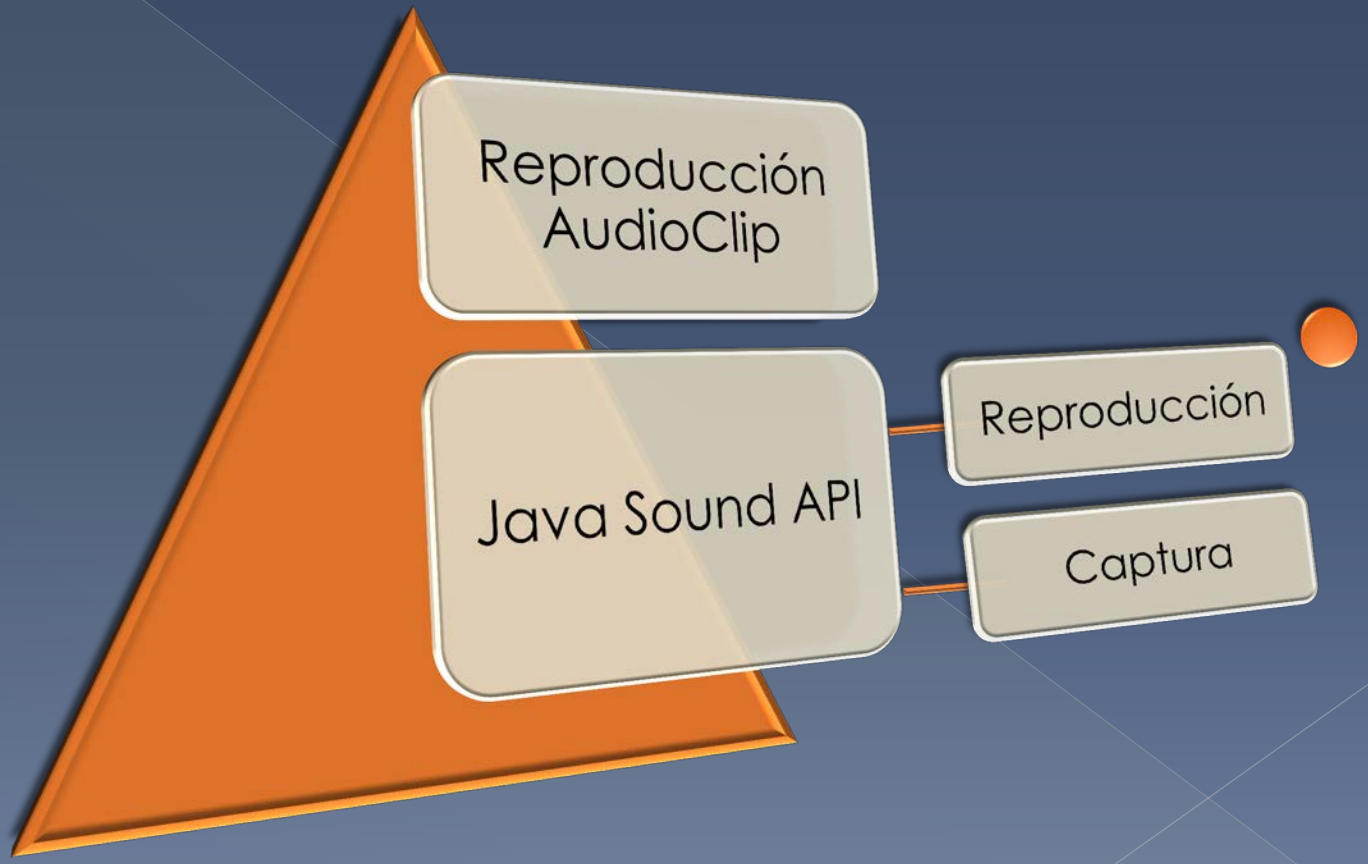
Formatos | Mezcladores | Líneas

Java Sound API

Resumen de clases

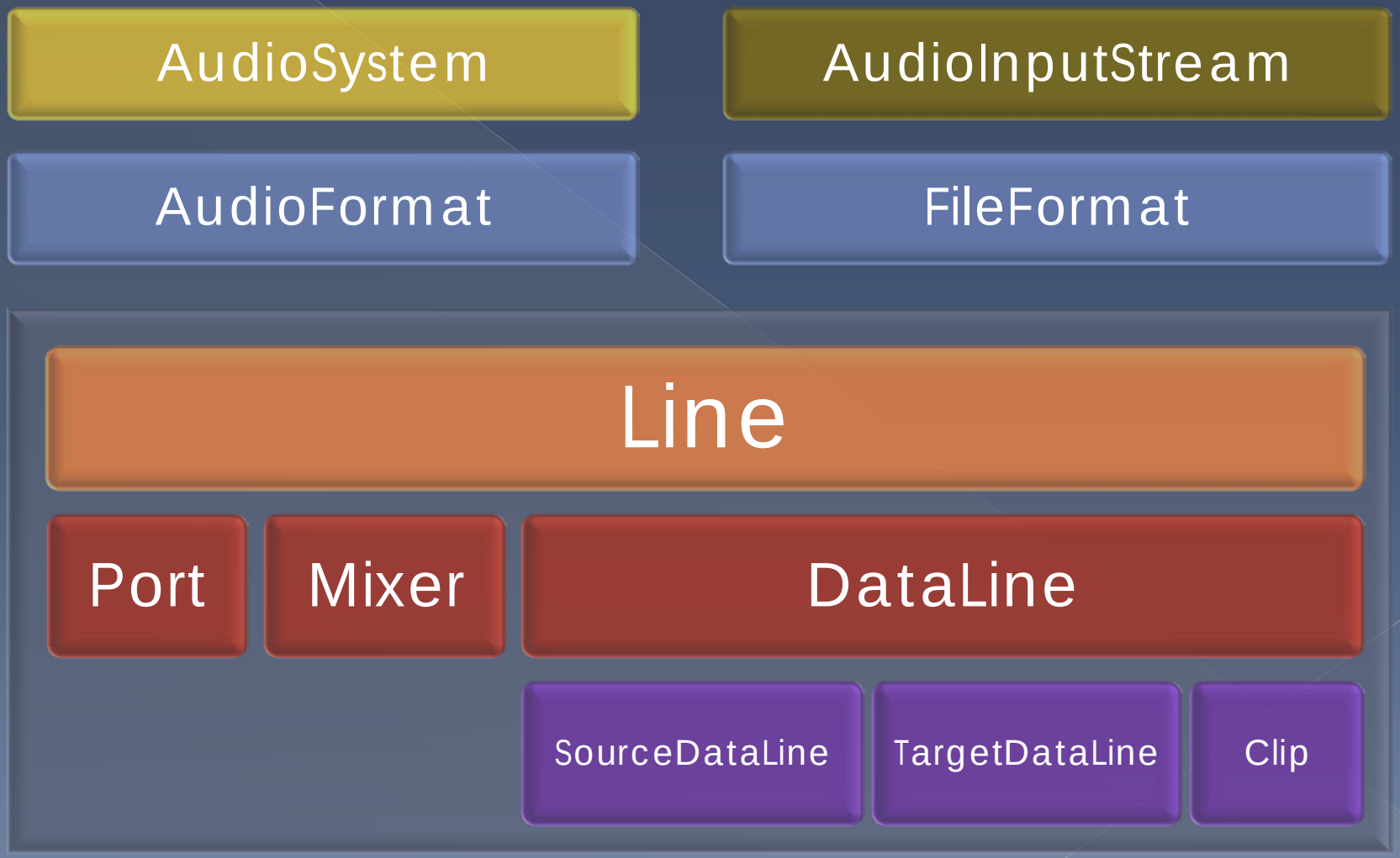


Índice


Reproducción AudioClip
Java Sound API
Reproducción
Captura

Java Sound API

Resumen de clases



Java Sound API

Resumen de clases

Reproducción

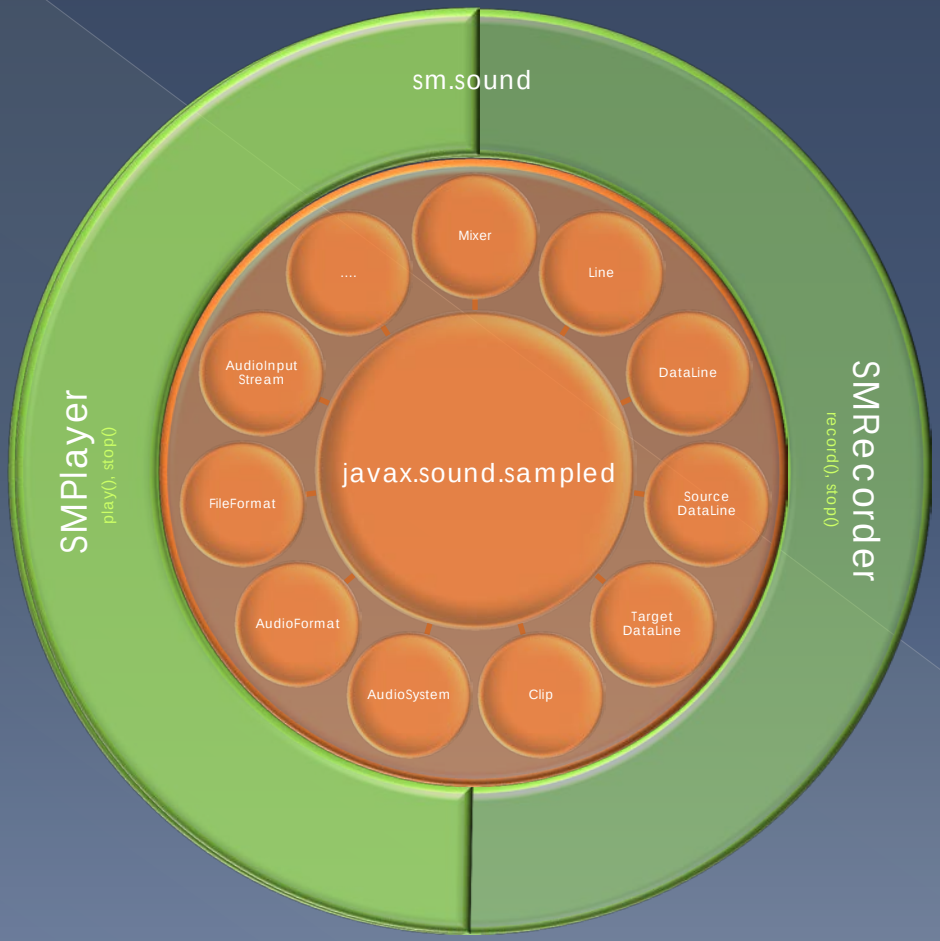
Clip

```
Clip sound;
File f=....;
LineListener lineListener=...;

public void play() {
      try{
            sound = AudioSystem.getClip();
            sound.addLineListener(lineListener);
            sound.open(AudioSystem.getAudioInputStream(f));
            sound.start();
      } catch (Exception e) {
           System.err.println(e);
      }
}

public void stop(){
      sound.stop();
}
```

Provee de un mezclador por defecto (internamente hace llamadas a getMixer y getLine)

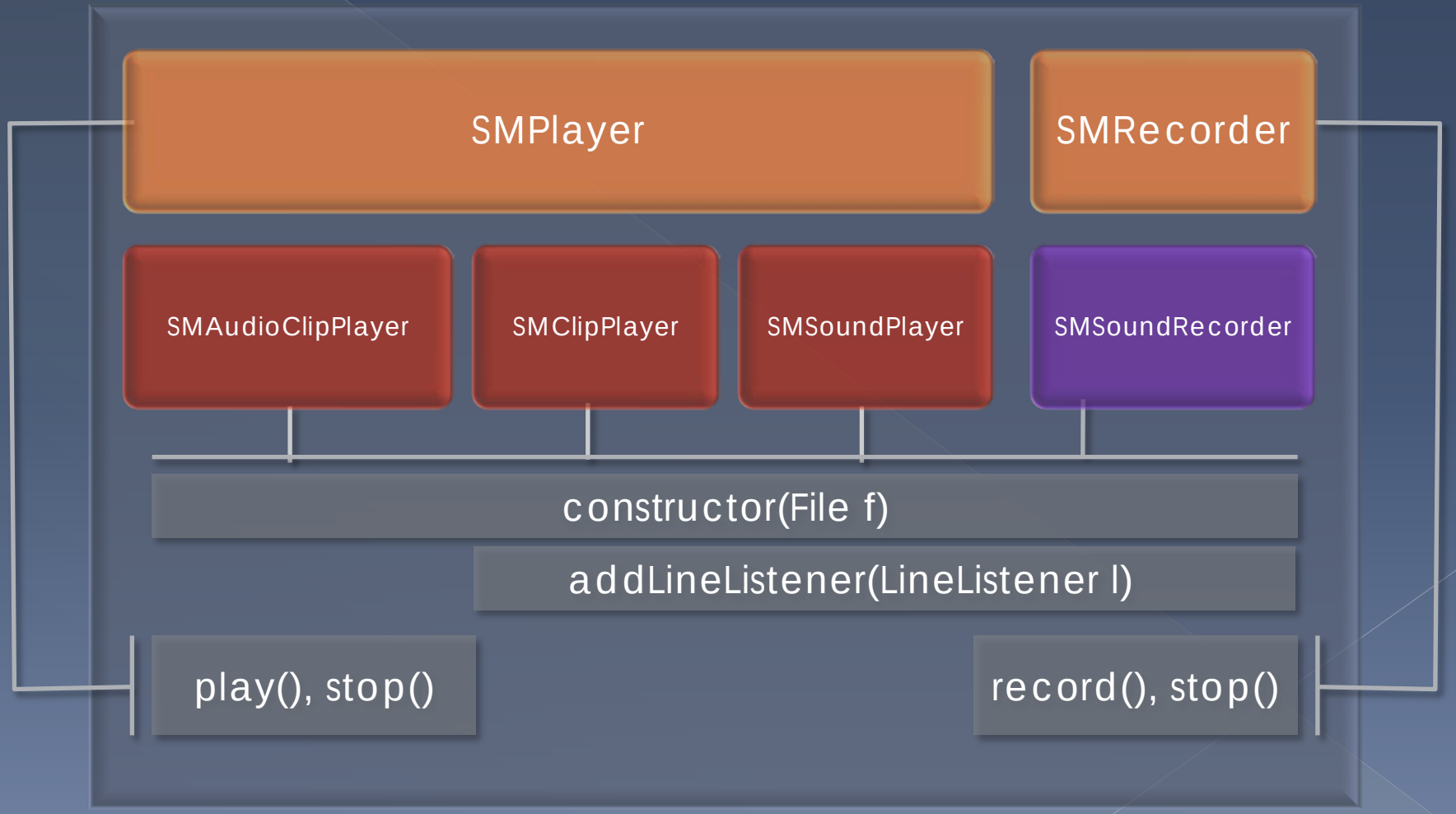
Reproducción

Paquete sm.sound



Reproducción

Paquete sm.sound



Reproducción

Paquete sm.sound

Reproducción

Clip en sm.sound

```
File f=....;


SMClipPlayer player = new SMClipPlayer(f);
If(player!=null) player.play();
```

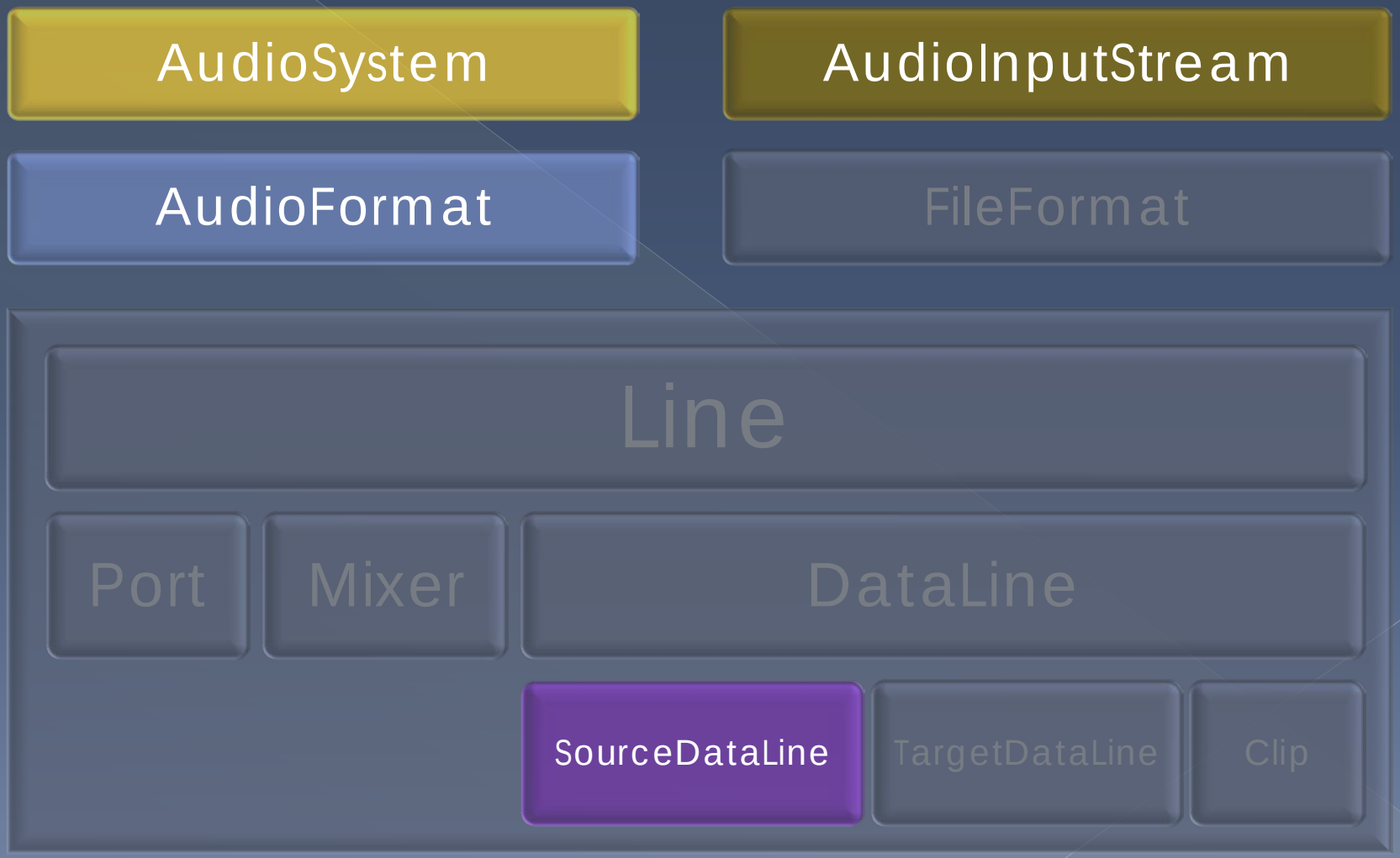
Reproducción

Clip en sm.sound

```
File f=....;
LineListener lineListener=...;

SMClipPlayer player = new SMClipPlayer(f);
If(player!=null) {
    player.addLineListener(lineListener);
    player.play();
}
```

Java Sound API

Resumen de clases



Reproducción

SourceDataLine

```
SourceDataLine line;
File soundFile=....;
LineListener lineListener=...;

public void startPlay() {
```

Formato de audio

```
    AudioInputStream audioInputStream = null;
    try{
          audioInputStream = AudioSystem.getAudioInputStream(soundFile);
    } catch (Exception e) {
         System.err.println(e);
    }
    AudioFormat audioFormat = audioInputStream.getFormat();
```

Línea

```
    DataLine.Info info = new DataLine.Info(SourceDataLine.class,audioFormat);
    try {
          line = (SourceDataLine) AudioSystem.getLine(info);
          line.addLineListener(lineListener);
          line.open(audioFormat);
    } catch (Exception e){
        System.err.println(e);
    }
```

Continúa

Reproducción

SourceDataLine

```
        line.start();
        int nBytesRead = 0;
        byte[] abData = new byte[128000];
        while(nBytesRead != -1) {
           try {
                 nBytesRead = audioInputStream.read(abData,0,abData.length);
                 if (nBytesRead >= 0) line.write(abData, 0, nBytesRead);
           } catch (IOException e){
              System.err.println(e);
           }
        }
        line.close();
    }

    public void stop(){
        line.stop();
        line.close();
    }
```

Lectura del stream
Escritura en la línea

Reproducción

Paquete sm.sound

Reproducción

SourceDataLine en sm.sound

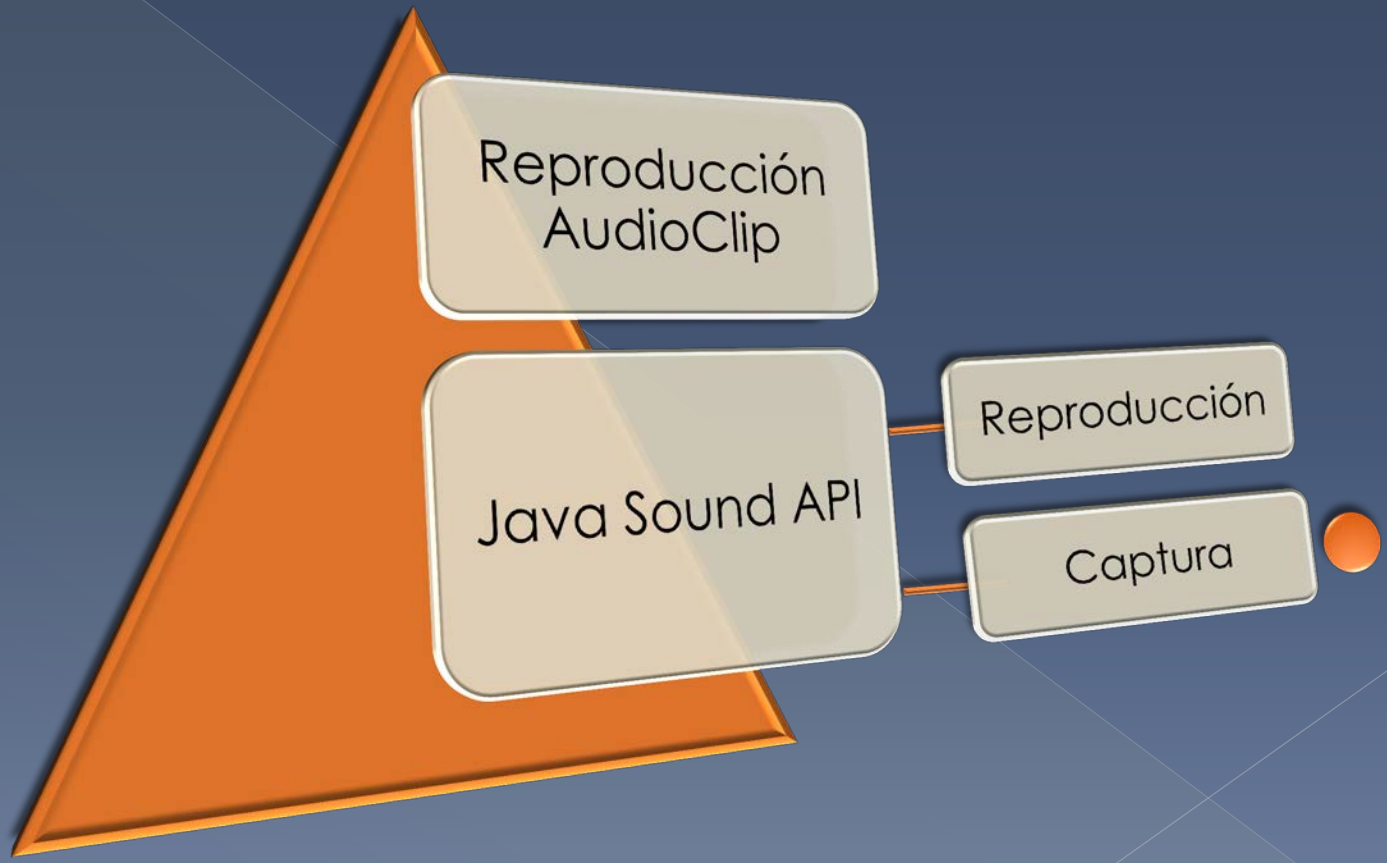
```
File f=....;


SMSoundPlayer player = new SMSoundPlayer(f);

If(player!=null) player.play();
```

Reproducción

SourceDataLine en sm.sound

```
File f=....;

LineListener lineListener=...;


SMSoundPlayer player = new SMSoundPlayer (f);

If(player!=null) {

    player.addLineListener(lineListener);

    player.play();

}
```

Índice


Reproducción AudioClip
Java Sound API
Reproducción
Captura

Java Sound API

Resumen de clases

Captura

TargetDataLine

```
TargetDataLine line;
File f=....;
public void startRecord() {
    AudioFormat audioFormat = new AudioFormat(AudioFormat.Encoding.PCM_SIGNED,
                                              44100.0f,16,2,4,44100.0f,false);
    DataLine.Info info = new DataLine.Info(TargetDataLine.class,audioFormat);
    try {
          line = (TargetDataLine) AudioSystem.getLine(info)
          line.addLineListener(lineListener);
          line.open(audioFormat);
     } catch (Exception e){
        System.err.println("SoundRecorder: "+e);
     }
     AudioFileFormat.Type targetType = AudioFileFormat.Type.WAVE;
     AudioInputStream audioInputStream = new AudioInputStream(line);
     line.start();
     try {
        AudioSystem.write(audioInputStream,targetType,soundFile);
     }
     catch (IOException e){
        System.err.println("SoundRecorder: "+e);
     }
}
```

Formato de audio

Línea

Formato de fichero

Escritura en la línea

Captura

TargetDataLine

```
public void stop() {
        line.stop();
        line.close();
}


public void record(){
       Thread thread = new Thread(){
           public void run() {
              startRecord();
           }
        };
        thread.start(); // Lanzamos la hebra
}
```

Captura

Paquete sm.sound

Captura

TargetDataLine en sm.sound

```
File f=....;


SMSoundRecorder rec = new SMSoundRecorder(f);

If(player!=null) rec.record();
```

Captura

TargetDataLine en sm.sound

```
File f=....;

LineListener lineListener=...;


SMSoundRecorder  rec = new SMSoundRecorder(f);

If(rec!=null) {

    rec.addLineListener(lineListener);

    rec.record();

}
```

Enlaces de interés

- Tutorial de Java Sound API:

  http://docs.oracle.com/javase/tutorial/sound/index.html

- Códigos ejemplo:

  http://www.jsresources.org/

JAVA: Sonido