OSUMTuto - Java Mp3 Player

By Jose Vzqz ; jose.v@sun.com ; vzqz_@msn.com

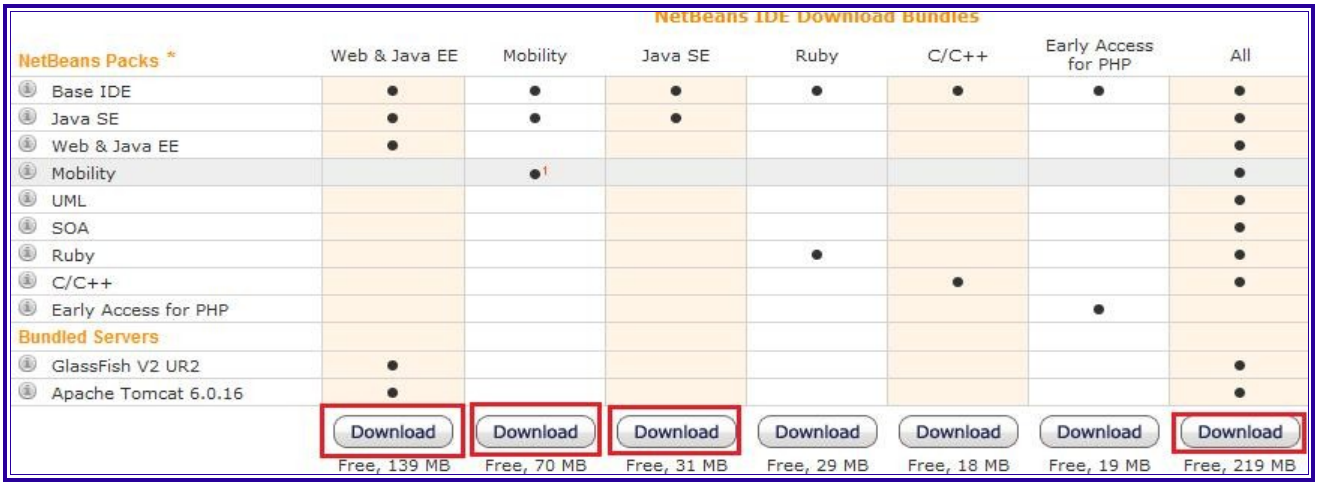
Bueno, pues sucede que ayer me encontraba haciendo una aplicación Web en Java cuando me surgió una duda que nada tenía que ver con lo que estaba haciendo, dije: Y si mi aplicación tocara música mientras está ejecutándose ? Entonces me dí a la tarea de buscar alguna manera de hacer esto, y este tutorial es el resultado de esa búsqueda.

El objetivo entonces es el siguiente:

> Lograr reproducir un mp3 en Java de una manera fácil y que nos permita seguir haciendo cualquier otra rutina mientras se reproduce el archivo.

Bueno, primero lo primero, Qué se necesita para este tuto?

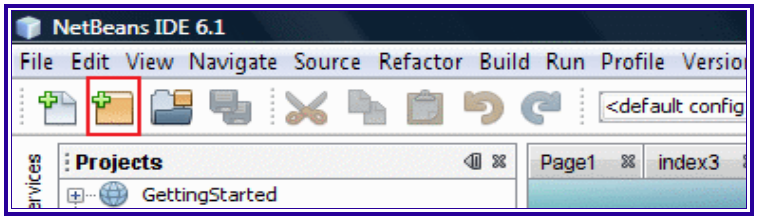
- Netbeans IDE 6.1 descargar aqui



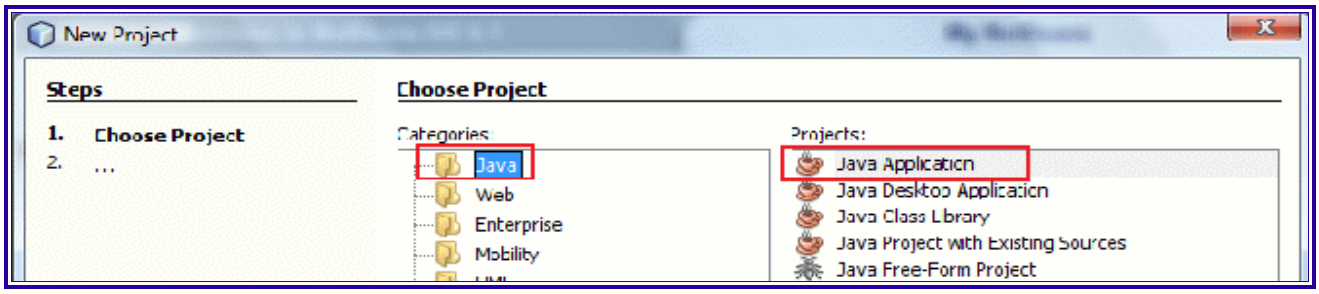
Cualquier opción que venga con el Base IDE y Java SE funcionarán para este tutorial.

- Librería Jlayer descargar aqui

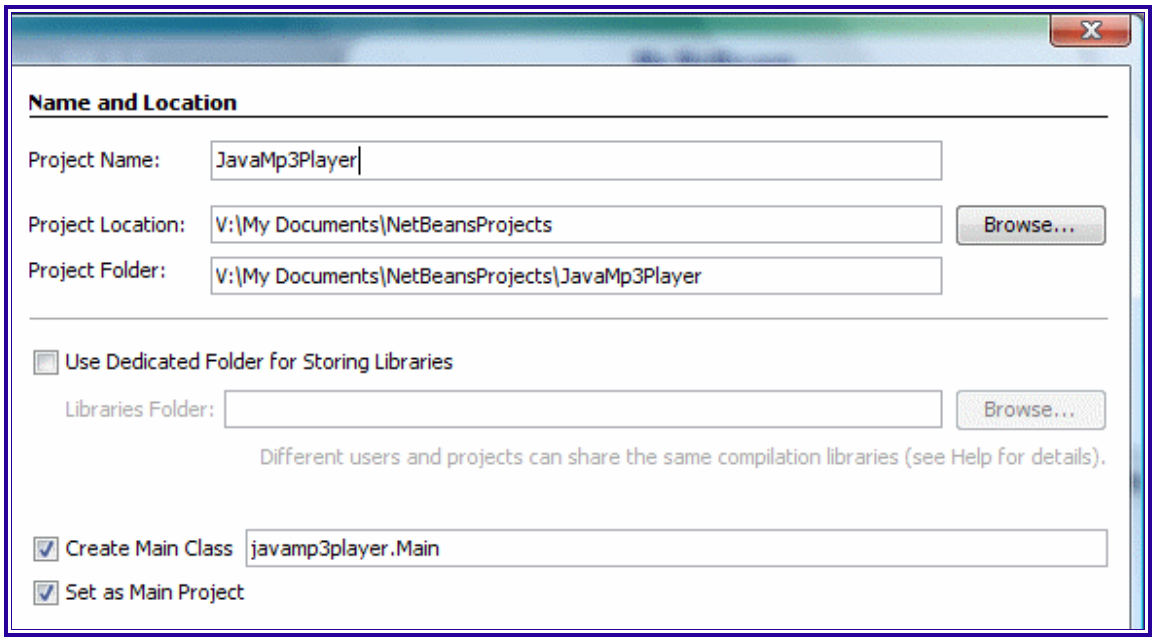
Listo, eso lo único que necesitamos, después de instalar Netbeans abrimos el IDE y generamos un nuevo proyecto:



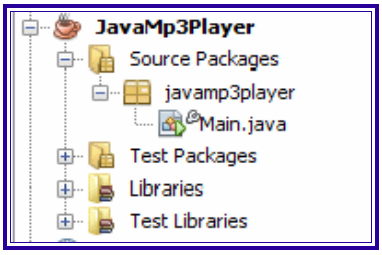
En la primer pantalla dejamos las opciones por default como muestra la imágen:



En la segunda pantalla le ponemos nombre a nuestro proyecto, en este caso usaremos el nombre: JavaMp3Player, las demás opciones las dejamos por default, y finalizamos el asistente.

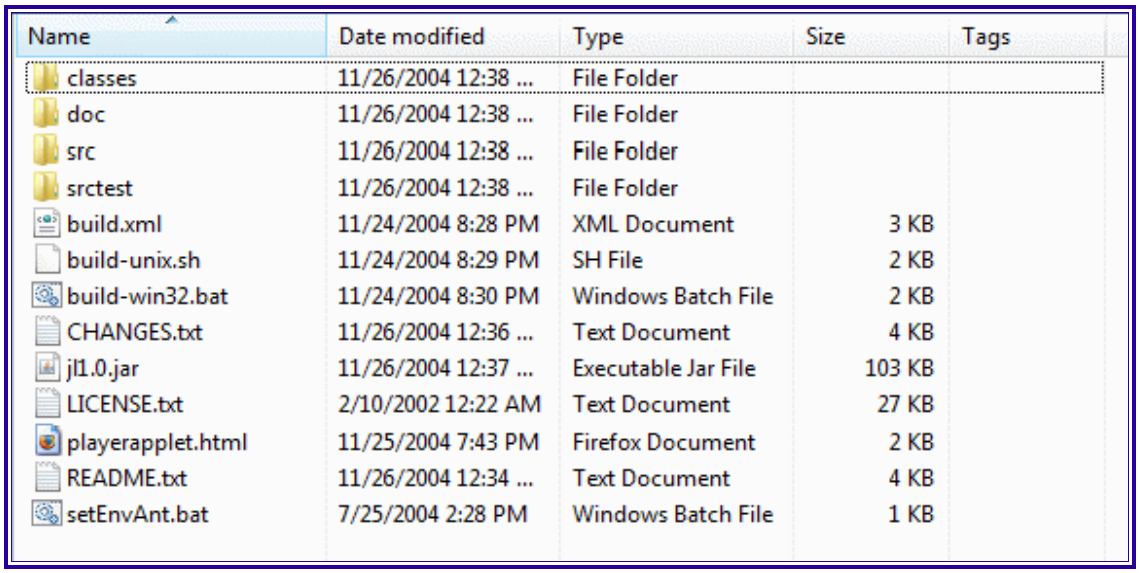


En la parte izquierda de nuestro IDE está la venta de proyectos, ahí debió aparecer nuestro nuevo proyecto de la siguiente manera:



Si por alguna razón no esta abierta esta ventana podemos verla de forma manual presionando Ctrl + F1

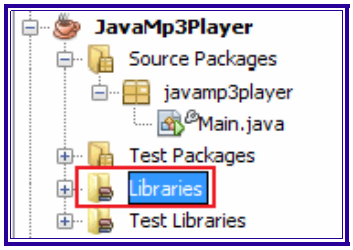
Una vez creado el proyecto debemos agregar la librería Jlayer, para agregarla a nuestro proyecto en Netbeans necesitamos el archivo que bajamos de la página javazoom.net al principio del tutorial. Descomprimimos el archivo en cualquier lugar y veremos el contenido de la carpeta:

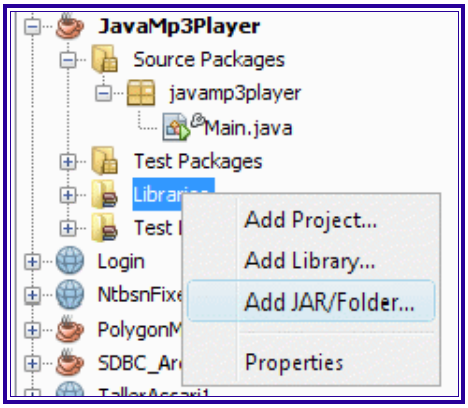


Name	Date modified	Type	Size	Tags
classes	11/26/2004 12:38 ...	File Folder		
doc	11/26/2004 12:38 ...	File Folder		
src	11/26/2004 12:38 ...	File Folder		
srctest	11/26/2004 12:38 ...	File Folder		
build.xml	11/24/2004 8:28 PM	XML Document	3 KB	
build-unix.sh	11/24/2004 8:29 PM	SH File	2 KB	
build-win32.bat	11/24/2004 8:30 PM	Windows Batch File	2 KB	
CHANGES.txt	11/26/2004 12:36 ...	Text Document	4 KB	
jl1.0.jar	11/26/2004 12:37 ...	Executable Jar File	103 KB	
LICENSE.txt	2/10/2002 12:22 AM	Text Document	27 KB	
playerapplet.html	11/25/2004 7:43 PM	Firefox Document	2 KB	
README.txt	11/26/2004 12:34 ...	Text Document	4 KB	
setEnvAnt.bat	7/25/2004 2:28 PM	Windows Batch File	1 KB	

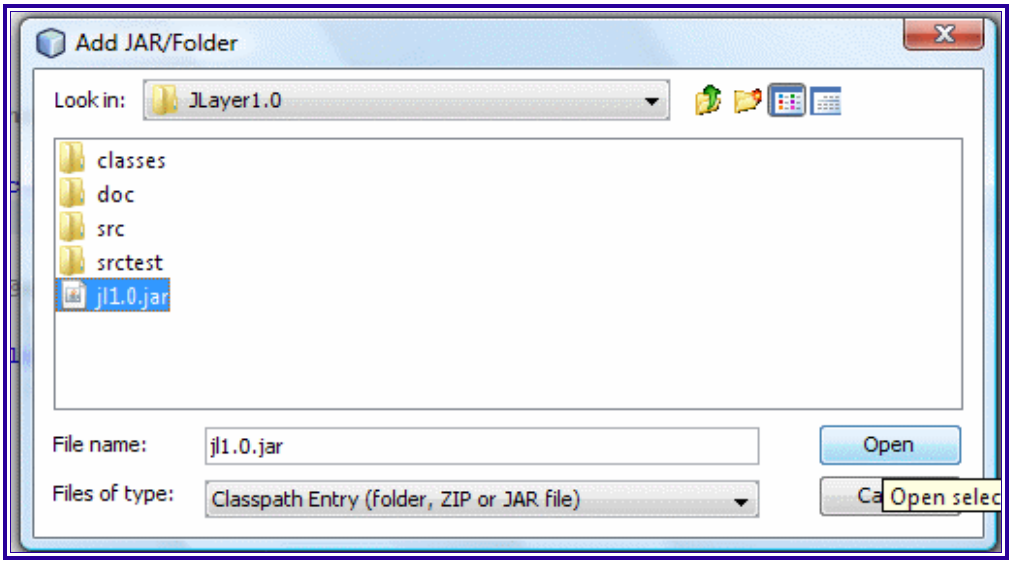
El único archivo que necesitamos es el jl1.0.jar que es donde están las clases ya compiladas del reproductor de mp3. Como podemos ver junto con la libreria también descargamos el código fuente y la documentación.

Una vez descomprida la carpeta procedemos a agregarla a nuestro proyecto en Netbeans.
Hacemos clic derecho sobre la parte de Libraries en nuestro proyecto.





Y seleccionamos Add JAR/FOLDER... esto nos lleva a una pantalla donde tenemos que buscar el directorio donde descomprimimos el archivo jl1.0.jar



Una vez agregado debe aparecer bajo la sección de Libraries del proyecto:

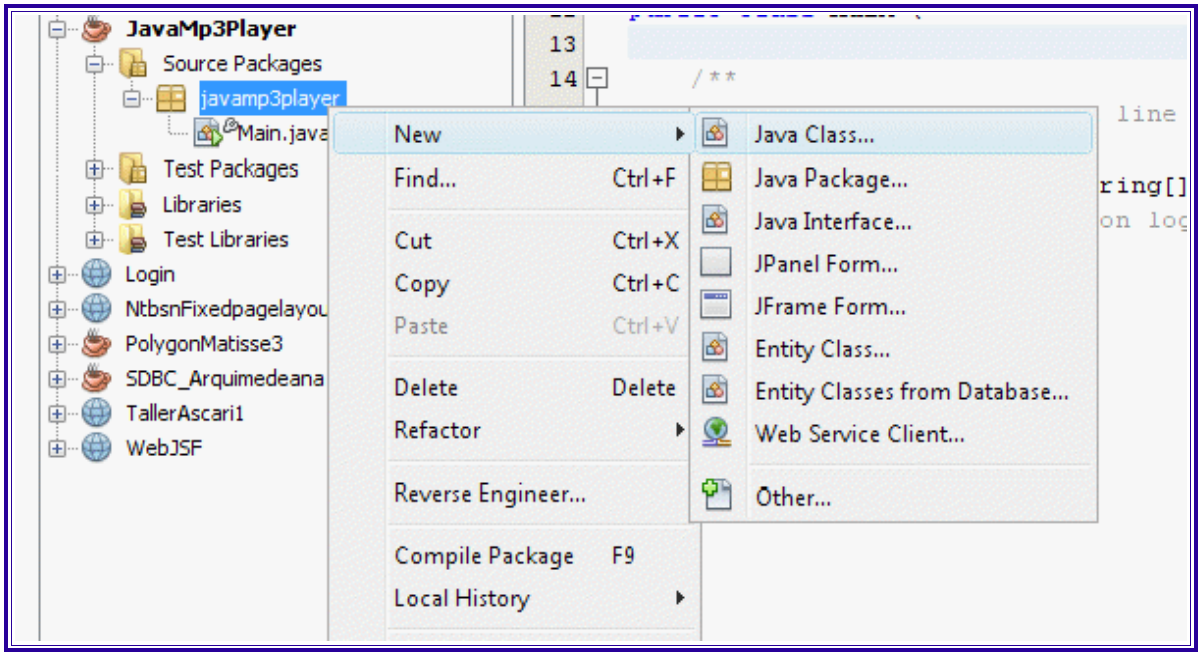
Ok, ahora tenemos todo listo para empezar a programar nuestro reproductor, a continuación explicaré como lo haremos. Para nuestro objetivo necesitamos 2 clases:

- Clase de control
- Clase de prueba

La clase de Control la necesitamos para especificar las funciones de nuestro reproductor. Esta clase llevará por nombre : MP3.java
La clase de prueba es una clase ejecutable que hace uso de la clase de control para probar su funconalidad. Esta clase llevará por nombre: Main.java

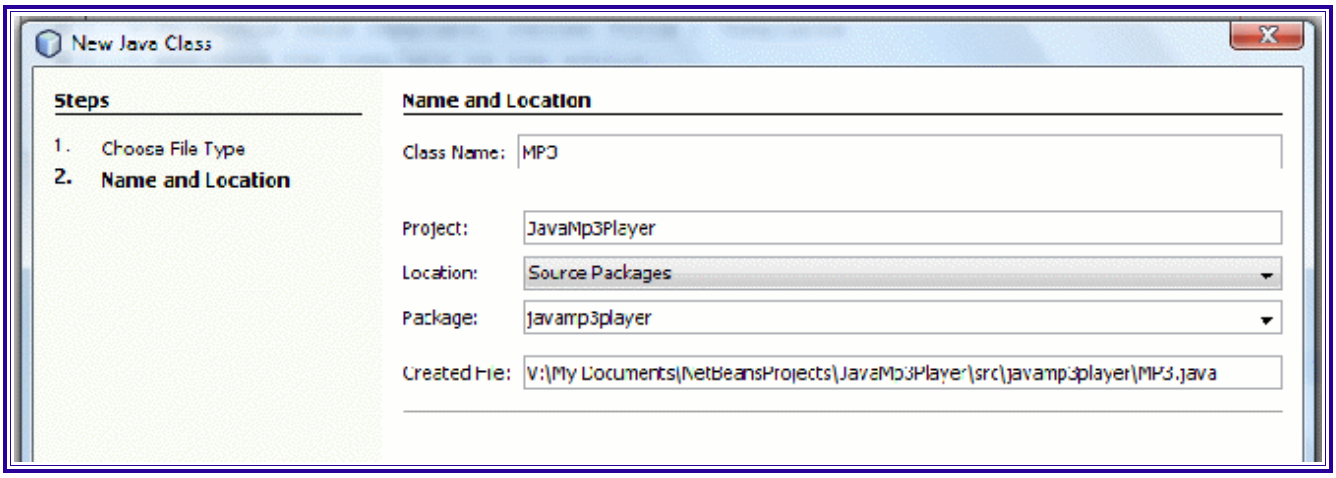
Empezemos por definir nuestra clase de control.

Sobre nuestro proyecto expandimos la parte de “Source Packages” y hacemos clic derecho sobre el paquete javamp3player, seleccionamos New > Java Class...



Esto nos llevará al asistente para generar una nueva clase. En el asistente ponemos como nombre de la clase: MP3 y lo demás lo dejamos en default, finalizamos el asistente.

Nota: Siempre es buena idea empezar el nombre de una Clase con mayúscula



Esto nos crea la clase llamada MP3.java y nos abre una ventana en el editor.

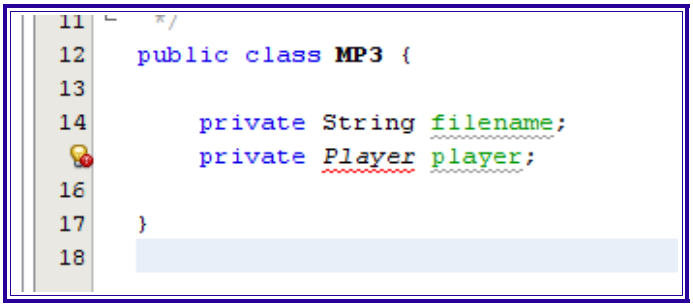
Declaramos una cadena y una variable tipo Player de la siguiente manera:

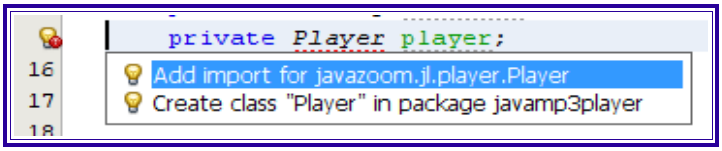
```
public class MP3 {

private String filename;
private Player player;

}
```

Como podemos notar Netbeans nos resalta un error sobre Player, esto significa que no puede encontrar alguna Clase con este nombre, la clase Player esta contendia dentro la librería jl1.0.jar por lo tanto tenemos que decirle a nuestro IDE en dónde la puede encontrar, para esto hacemos clic sobre el foco amarillo que aparece al principio de la línea donde declaramos Player y seleccionamos la primera opción: “Add import for javazoom.jl.player.Player”.





Al hacer esta acción lo que hace el IDE es agregar el import necesario al principio del código:

```
import javazoom.jl.player.Player;
```

Por lo tanto nuestro código debe parecerse a esto:

```
package javamp3player;
import javazoom.jl.player.Player;

public class MP3 {

   private String filename;
   private Player player;

}
```

La cadena filename la usaremos para recibir el nombre del archivo mp3 a tocar, el objeto player lo usaremos para controlar nuestro mp3.

A continuación declaramos el constructor de la clase

```
public MP3(String archivo) {
      this.filename = archivo;
}
```

Este constructor recibe como argumento una cadena que se asigna a la propiedad filename para futuras referencias al archivo.

Ahora empezaremos a declarar los métodos con los que controlaremos nuestro player, empezamos con el método Close

```
en este método cerramos la conexión al player si es que se le ha asignado alguna

public void close()             {
        if (player != null) {
            player.close();
        }
}
```

El siguiente método es el más “complicado” de nuestra clase de control, es el método Play, aquí leemos el archivo y se lo pasamos al objeto Player para que lo reproduzca.

```
public void play() {
        try {
            FileInputStream fis = new FileInputStream(filename);
            BufferedInputStream bis = new BufferedInputStream(fis);
            player = new Player(bis);
        } catch (Exception e) {
            System.out.println("Problem playing file " + filename);
            System.out.println(e);
        }
        // run in new thread to play in background
        new Thread() {

            public void run() {
                try {
                    player.play();
                } catch (Exception e) {
                    System.out.println(e);
                }
            }
        }.start();
    }
```

En la primera parte de la función declaramos un objeto de tipo FileInputStream donde como parámetro le pasamos el archivo que especificamos. Después el fis se lo pasamos a otro objeto igual de lectura llamado BufferedInputStream que se encarga de hacer la lectura con Buffer, ya que es un archivo pesado. Los bloques try-catch nos avisan si ha ocurrido un error de lectura con nuestro archivo.

Después el método play crea un hilo de proceso para reproducir el archivo en un proceso aparte, de manera que nuestro programa siga su curso, nuevamente este proceso lo encierra en un bloque try-catch para que nos avise si ha ocurrido algún error a la hora de reproducir el archivo.

Al construir el código del método play Netbeans nos recomendará agregar los siguientes imports:

```
import java.io.BufferedInputStream;
import java.io.FileInputStream;
```

Ya solo faltan dos métodos que agregar, uno es el método para detectar si el archivo está tocando.

```
public boolean isComplete() {
        return player.isComplete();
}
```

Y otro método para saber cuantos segundos han pasado desde que se inició la canción.

```
public int getPos(){
        return player.getPosition();
}
```

Ok, hasta aquí programamos la clase de control deberán tener algo como esto:

```
package javamp3player;
import java.io.BufferedInputStream;
import java.io.FileInputStream;
import javazoom.jl.player.Player;

public class MP3 {

    private String filename;
    private Player player;

    public MP3(String archivo) {
        this.filename = archivo;
    }

    public void close() {
        if (player != null) {
            player.close();
        }
    }

    public void play() {
        try {
            FileInputStream fis = new FileInputStream(filename);
            BufferedInputStream bis = new BufferedInputStream(fis);
            player = new Player(bis);
        } catch (Exception e) {
            System.out.println("Problem playing file " + filename);
            System.out.println(e);
        }

        new Thread() {

            public void run() {
                try {
                    player.play();
                } catch (Exception e) {
                    System.out.println(e);
                }
            }
        }.start();
    }

    public boolean isComplete() {
        return player.isComplete();
    }

    public int getPos() {
```

```
        return player.getPosition();
    }
}
```

Ahora agregaremos el código a la clase de prueba o Main.java

Por default Netbeans nos crea la clase Main con el siguiente contenido:

```
package javamp3player;

public class Main {
    public static void main(String[] args) {
    }
}
```

El código que vamos a agregar va dentro del método main, este método es el que reconoce la máquina virtual de Java como el método principal del programa, como vemos recibe un parámetro llamdo args que es un arreglo de caracteres, esto nos es de utilidad cuando queremos pasar argumentos directamente a nuestro programa cuando lo iniciamos, esta funcionalidad la usaremos en nuestro código de la siguiente manera:

Nota: El siguiente código va dentro del método main.

Primero declaramos una cadena llamada filename y la igualamos al primer argumento que está en la posición cero de nuestro arreglo args.

```
String filename = args[0];
```

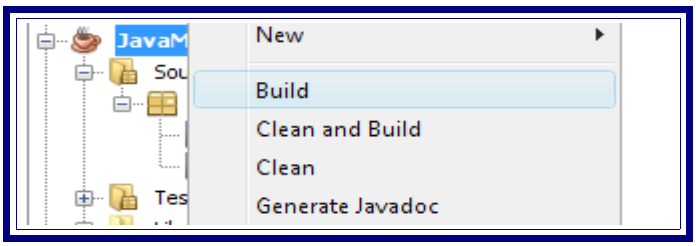
Después creamos un objeto de la clase MP3 llamado miMp3, lo inicializamos y mandamos llamar su método play.

```
MP3 miMp3 = new MP3(filename);
miMp3.play();
```

Nota: new MP3 (filename) manda llamar el constructor que programamos en la clase MP3

Hasta aquí nuestro programa es ahora funcional, para hacer una prueba hacemos lo siguiente:

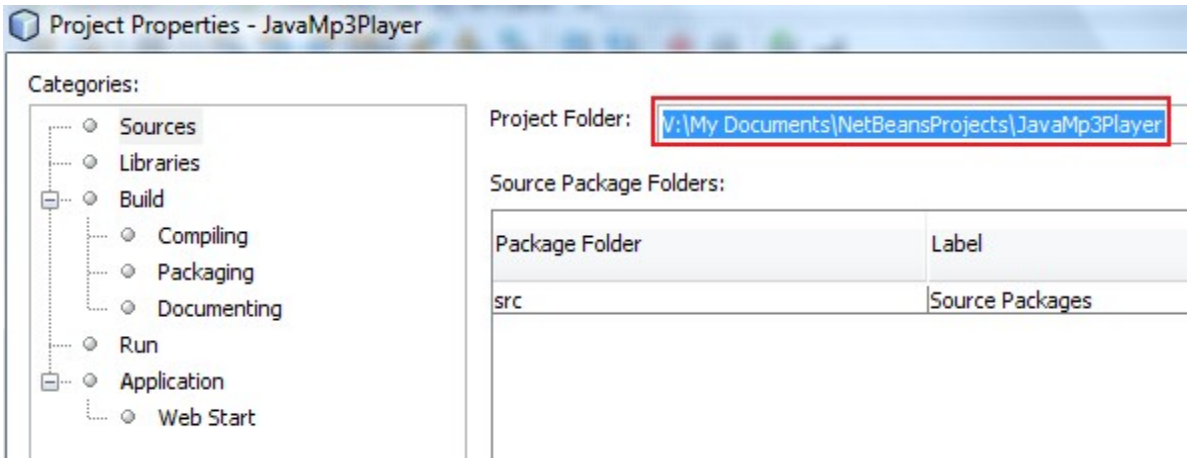
Necesitamos compilar nuestra clase, le damos clic derecho a nuestro proyecto y seleccionamos la opción build.



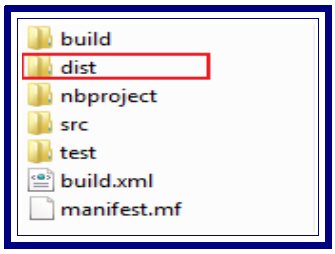
Esto nos generará una carpeta llamda “dist” en la carpeta de nuestro proyecto. Dentro de esta carepta se

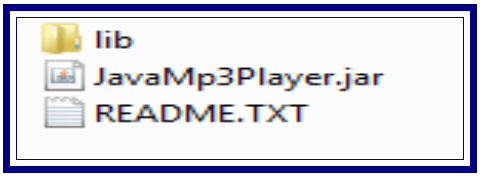
genera el archivo .jar que contiene nuestro programa.

Para ubicar la carpeta de nuestro proyecto damos clic derecho sobre el ícono de nuestro proyecto y seleccionamos “Propiedades”



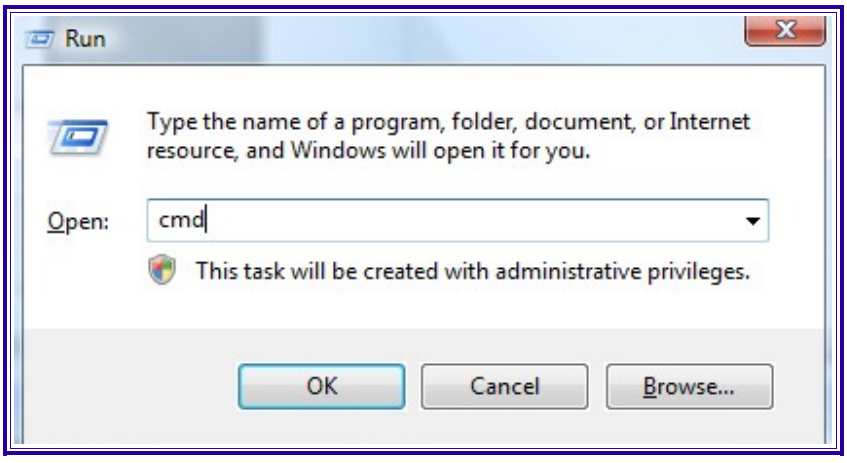
Una vez que tenemos el path del proyecto podemos ver la carpeta que generó Netbeans a la hora de hacerle el Build al proyecto.





Abrimos la carpeta y encontramos el archivo .jar

Abrimos una consola...



Y navegamos a la carpeta dist de nuestro proyecto:

Una vez ahí ejecutamos el siguiente comando:

```
java -jar JavaMp3Player.jar lemon.mp3
```

Donde lemon.mp3 es el archivo que debe estar ubicado en la misma carpeta que el archivo JavaMp3Player.jar.

Una vez ejecutado el comando debemos ser capaces de escuchar la canción!

Ahora pongamos un poco de valor extra en la aplicación, hagamos un ciclo en el cual nos muestre los segundos que ha avanzado la canción y que al final nos avise cuando se haya detenido. Para esto agregamos el siguiente código después de la última línea que escribimos en el bloque main:

```
while(!miMp3.isComplete())
{
   System.out.println("Segundos: "+miMp3.getPos()/1000);
}

System.out.println("Fin de reproduccion");
miMp3.close();
System.out.println("Archivo cerrado");
JOptionPane.showMessageDialog(null,"La canción Finalizó!!");
```

El while hace uso del método isComplete() que programamos en la clase MP3, lo que hace es imprimir en pantalla los segundos usando el método getPos() que también programamos en la clase MP3. Cuando detecte que la reproducción ha terminado, manda un mensaje al usuario.

Nota: para usar el MessageDialog hay que agregar el siguiente import

```
import javax.swing.JOptionPane;
```

Listo, tenemos un programa en Java que reproduce un archivo MP3 !!

El código de nuestra clase Main.java debe parecerse a esto:

```
package javamp3player;
import javax.swing.JOptionPane;

public class Main {

   public static void main(String[] args) {

      String filename = args[0];
      MP3 miMp3 = new MP3(filename);
      miMp3.play();

      while(!miMp3.isComplete())
      {
         System.out.println("Segundos: "+miMp3.getPos()/1000);
      }

      System.out.println("Fin de reproduccion");
      miMp3.close();
      System.out.println("Archivo cerrado");
      JOptionPane.showMessageDialog(null,"La canción Finalizó!!");

   }

}
```

Usando este nucleo podemos programar algo más profesional usando un entorno gráfico pero ese ya es otro Tutorial que deberá ser escrito en otra ocasión.

Fuentes:

http://www.cs.princeton.edu/introcs/faq/mp3/mp3.html

http://www.javazoom.net/javalayer/docs/docs1.0/index.html

http://www.javazoom.net/javalayer/sources.html

Agradecimientos:

Kevin Wayne

Ruben Zafra

Testers

Josué Chimal

Jose Vzqz - Sun Microsystems
osum.sun.com/group/uaeh

OSUMTuto - Java Mp3 Player
Tuto creado con OpenOffice3
Versión 1.2 Fecha 19.10.08