



Estudiante

Elvis Lizandro Aguilar Tax

Carnet

201930304

Manual de Usuario

Descripción del programa

El sistema consta de una arquitectura de cliente servidor, para lo que consta de dos partes: un módulo servidor, es quien recibe como entrada un lenguaje de alto nivel describiendo la creación de pistas y listas de reproducción, almacenamiento de canciones, así como otras funciones. El módulo consumidor/cliente, reproduce las pistas y tendrá una forma de crear sus propias canciones también.

La manera de comunicarse de ambos módulos es por medio de sockets.

El módulo servidor deberá ser capaz de crear nuevas pistas, además de ordenarlas por listas de reproducción, las cuales podrán ser reproducidas también por el servidor, mostrando una gráfica que permita identificar la frecuencia de cada nota contenida en la pista.

El módulo cliente se encargará de solicitar y reproducir cada pista desde cada lista de reproducción definida en el módulo servidor. Cabe mencionar que este módulo permite la creación de canciones a través de un apartado especial de la aplicación que contiene un teclado musical integrado.

Lenguaje de definición de pistas

Comentarios de una línea

```
>>comentario de una sola línea
```

Comentarios de bloque

Estos deben empezar con un signo de “menor que” y un guión y termina con un guión y un signo de “mayor que”:

```
<- comentario de varias líneas  
Segunda línea  
Tercera línea  
...  
->
```

Pista

La sintaxis para la declaración de una pista es la siguiente, en donde debe de establecerse el nombre

de la pista y también si se extiende de otra pista. Extender de otra pista es opcional para cada pista que se vaya a crear.

```
Pista FragmentoCoro {  
    >> cuerpo de la pista  
    ...  
}
```

```
Pista Coro extiende Ritmo, FramgentoCoro {  
    >> Sentencia 1;  
    >> Sentencia 2;  
    >> Ciclo 1 {  
        >> Sentencia 1;  
        >> Sentencia 2;  
        >> Ciclo 1_1 {  
            >> Sentencia 1;  
            >> Sentencia 2;  
            ...  
        }  
        >> Sentencia 3;  
        ...  
    }  
    >> Sentencia n;  
}
```

Tipos de datos

Nombre	Descripción	Ejemplos	Observaciones
entero	Este tipo de dato acepta solamente números enteros	1, 50, 100, 255125	No tiene límite de tamaño
doble	Es un entero con decimal	1.2, 50.23, 0.34	Se maneja como regla general el manejo de 6 dígitos después del punto decimal

boolean	Admite valores de verdadero o falso, y variantes de los mismos	verdadero, falso, true, false. 1=verdadero, 0=falso	si se asigna los enteros 1 o 0 debe aceptar como verdadero o falso según el ejemplo
caracter	Solo admite un carácter por lo que no será posible ingresar cadenas enteras. Viene encerrado en comilla simple entre a-z, A-Z, -, _ : incluyendo símbolos y caracteres especiales #t, #n	'a', 'b', '#t', ' '), ""	En el caso de querer escribir comilla simple se escribirá primero # (caracter de escape) y después la comilla simple '#', además si se quiere escribir # se pondrá 2 veces "##", los caracteres especiales se escriben siempre '#t' y '#n'

cadena	Este es un conjunto de caracteres que pueden ir separados por espacios en blanco encerrados en comilla doble, los posibles caracteres a ingresar serán los mismos que los del tipo caracter solo que agregando a estos el espacio	"cadena1", "est##\$%o es una ca&/(de=na #n#n!!!"	En el caso de querer escribir comilla doble se escribirá primero # y después la comilla doble "ho#"la mundo #"" , además si se quiere escribir # se pondrá 2 veces"##", por ejemplo "ho#la mundo ##", los caracteres especiales se escriben "#t" y "#n".
--------	---	---	--

Operadores relacionales

Son los símbolos utilizados en expresiones booleanas, su finalidad es comparar expresiones entre sí

dando como resultado booleanos. En el lenguaje serán soportados los siguientes:

Nombre	Símbolo	Ejemplos	Descripción
Igualación	==	1==1, "hola"=="hola", 25.2555==80.051	Compara ambos valores y verifica si son iguales

Diferenciación	!=	1!=2, martes!=vari1	Compara ambos lados y verifica si son distintos
Mayor que	>	(5+5.5)>8.98	Compara ambos lados y verifica si el izquierdo es mayor que el derecho izquierdo.
Menor que	<	(5/(5+5))<(8*8)	Compara ambos lados y verifica si el derecho es mayor que el izquierdo
Mayor o igual que	>=	5-6>=4+6	Compara ambos lados y verifica si el izquierdo es mayor o igual que el derecho
Menor o igual que	<=	55+66<=44	Compara ambos lados y verifica si el derecho es mayor o igual que el izquierdo
Es nulo	!!	!!var1	Verifica si la variable fue declarada pero no le fue asignado algún valor inicial o una asignación posterior

Operadores Lógicos

Nombre	Símbolo	Evento	Ejemplo	Observaciones
AND	&&	Compara expresiones lógicas y si son ambas verdaderas entonces devuelve verdadero en otro caso retorna falso	(flag1) && ("hola" == "hola") RESULTADO ES FALSO	flag1 es falso
NAND	!&&	Compara expresiones lógicas y si son ambas verdaderas entonces devuelve falso en otro caso retorna verdadero	(flag2) && ("hola" == "hola") RESULTADO ES FALSO	flag1 es verdadero
OR		Compara expresiones lógicas y si al menos una es verdadera	(55.5<4) (!!bandera)	bandera es null

Aritmeticos

+,-,*,/,^

Declaración de variables

```
Pista Coro extiende Ritmo, FramgentoCoro {
    var entero numero1;
    Keep var cadena cad1,cad2,cad3,titulo,artista;
}
```

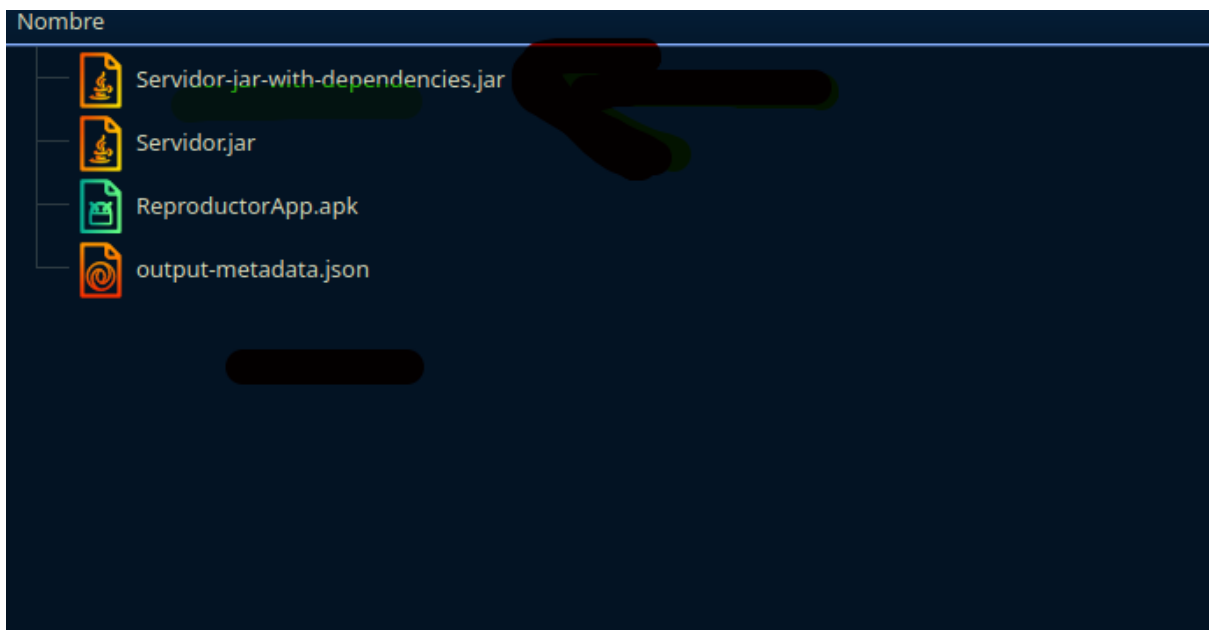
Condicional Si - sino si - sino

Switch

```
switch(variable) {  
    caso valor1:  
        >> Sentencias caso 1  
        salir  
    caso valor 2:  
        >> Sentencias caso 2  
        salir  
    caso valor 3:  
        >> Sentencias caso 3  
        salir  
    ...  
    default:  
        >> Sentencias default
```

etc.....

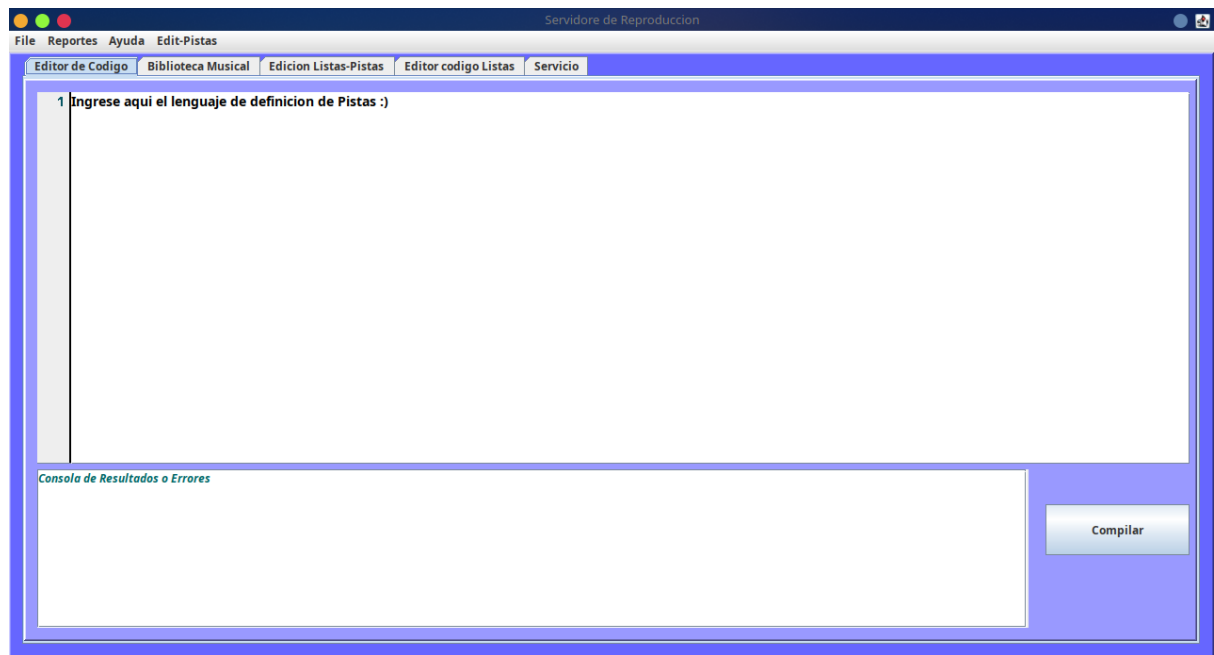
Manual de Uso



busque este ejecutable y en una consola o terminal ingrese el comando

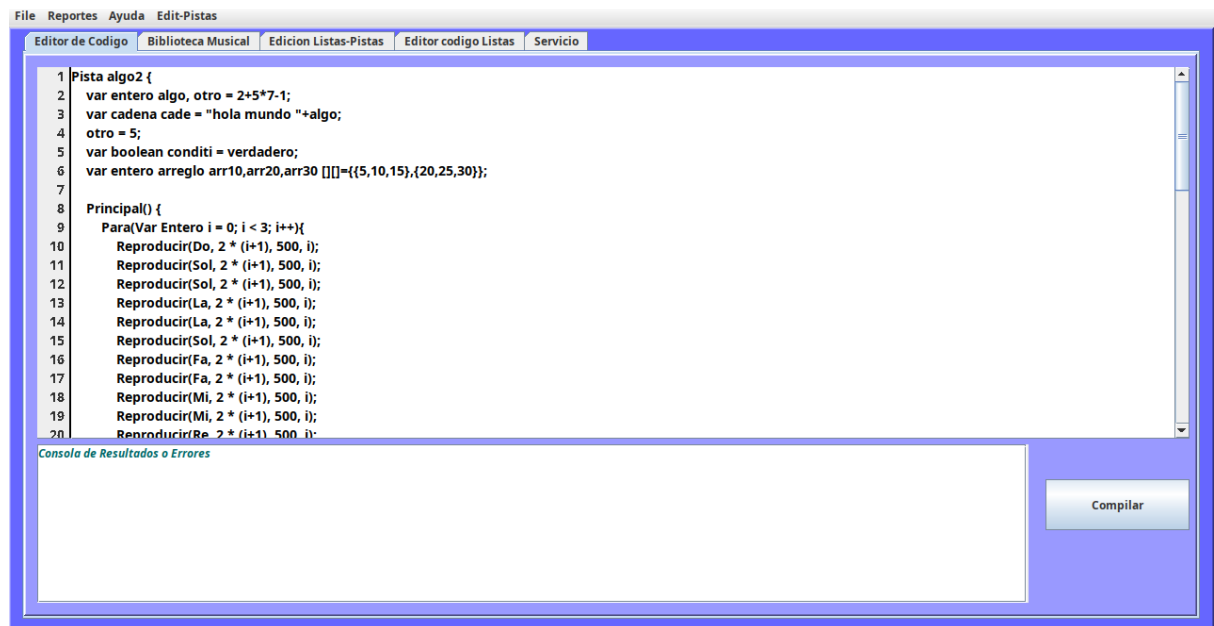
java -jar Servidor-jar-with-dependencies.jar

```
~/ReproductorMusicComp2Proyct1/ejecutables > main • java -jar Servidor-jar-with-dependencies.jar
```

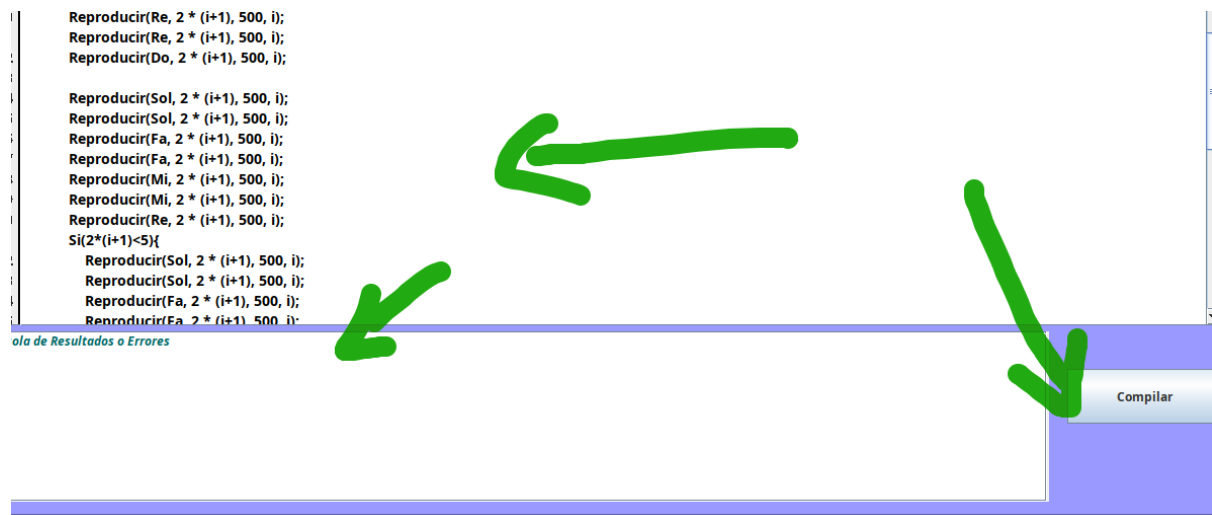


se le abrirá esta ventana

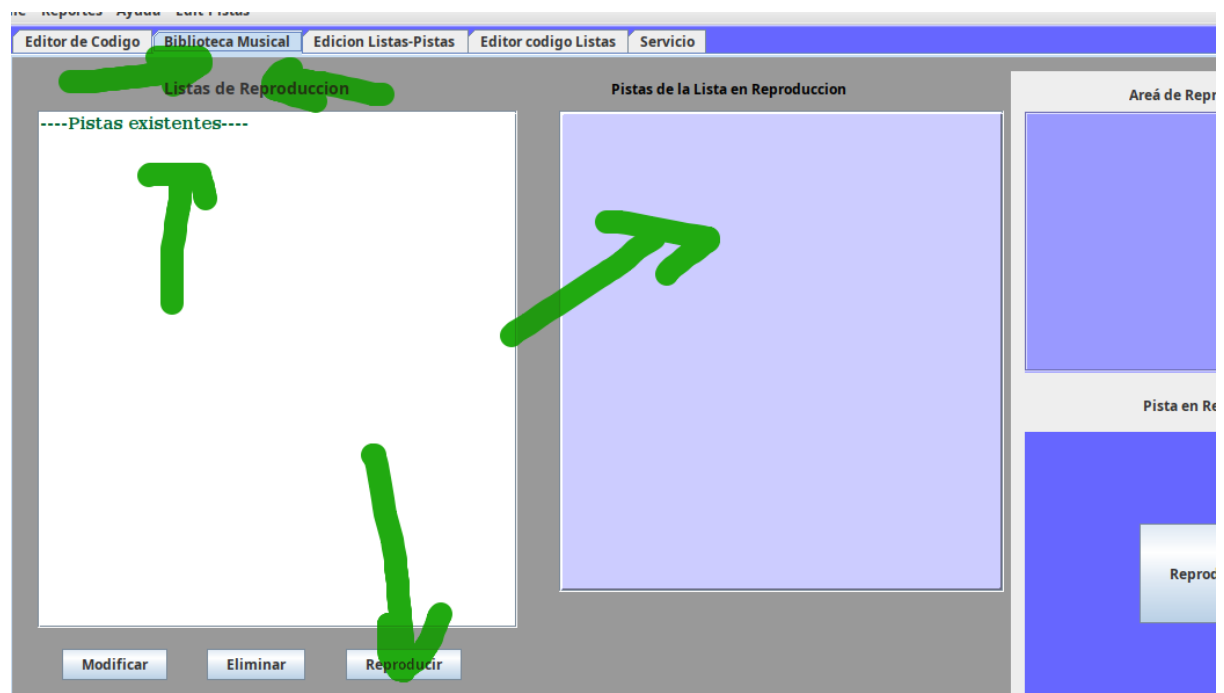
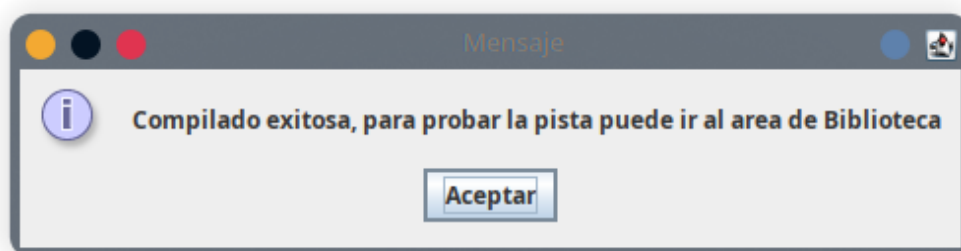
Ingresa un codigo ejecutable, como previamente se indico



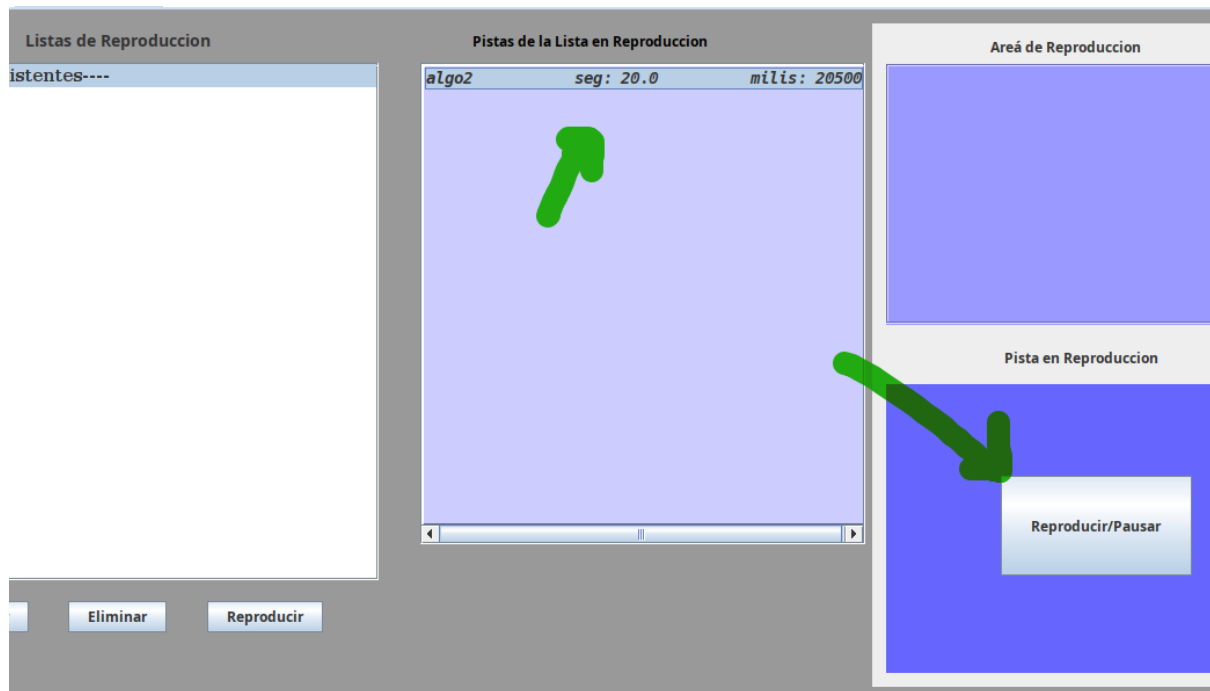
compile el codigo



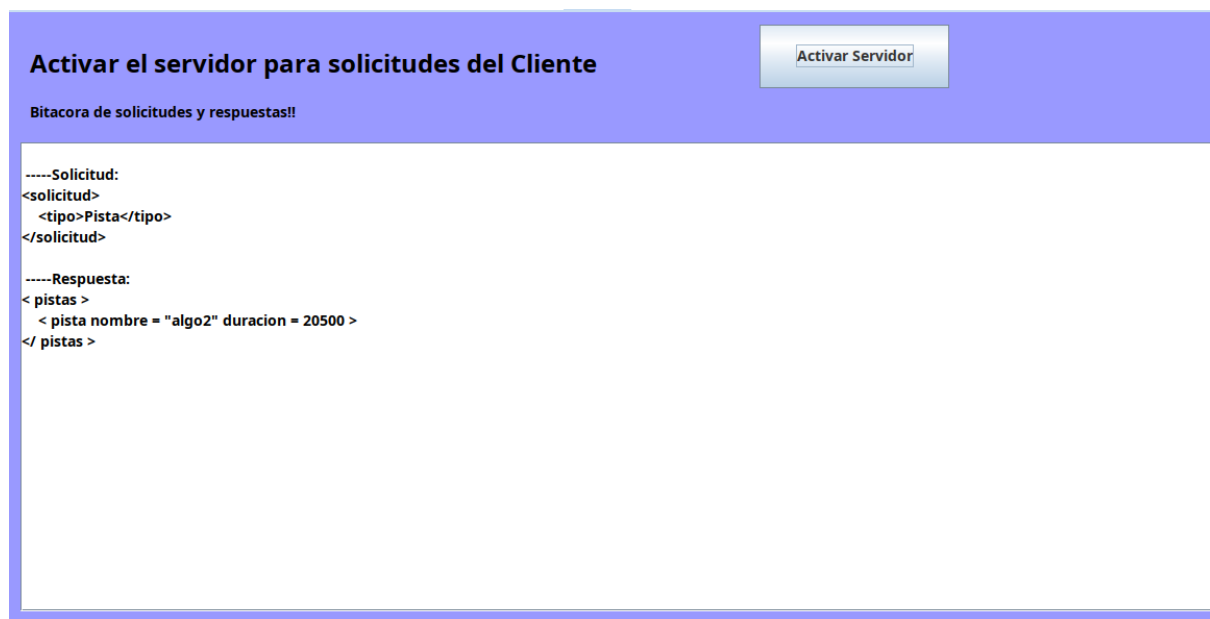
cuando el codigo es compilado de buena manera



dirigete a la biblioteca musical y reproduce la pista



Comunicacion entre cliente y servidor





Aplicacion Movil para solicitar
Pistas y listas de pistas al
servidor

Menu de Opciones

Solicitar Listas/
Pistas

Ver Reporete
Errores

Configurar
conexion

10:51 AM | 0.3kB/s | 23%

Configuracion de Conexion,
ingrese la ip de conexion

ejemplo: 192.168.0.110

Confirmar

Importante Solicitar la Lista que quieres Reproducir

Items

algoz



Reproducir/Pausar

Solicitud de Listas Y pistas

```
<solicitud>  
  <tipo>Pista</tipo>  
</solicitud>
```

Enviar
Solicitud

```
< pistas >  
  < pista nombre = "algo2" duracion = 20500  
>  
</ pistas >
```